



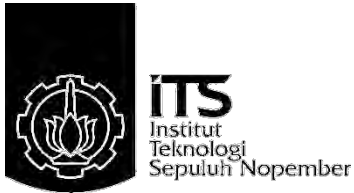
TUGAS AKHIR - TE141599

**PEMODELAN KURVA KARAKTERISTIK INVERSE
NON-STANDAR PADA RELE ARUS LEBIH DIGITAL
DENGAN METODE INTERPOLASI LAGRANGE**

Nurio Herlambang
NRP 2214105043

Dosen Pembimbing
Dr. Ir. Margo Pujiantara, M.T.
Dr. Eng. Ardyono Priyadi, S.T., M. Eng.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016



FINAL PROJECT - TE141599

**NON-STANDARD INVERSE CHARACTERISTIC CURVE
MODELING OF DIGITAL OVERCURRENT RELAY WITH
LAGRANGE INTERPOLATION METHODE**

Nurio Herlambang
NRP 2214105043

Advisor
Dr. Ir. Margo Pujiantara, MT.
Dr. Eng. Ardyono Priyadi, ST., M.Eng.

DEPARTMENT OF ELECTRICAL ENGINEERING
Faculty of Industrial Technology
Sepuluh Nopember Institute of Technology
Surabaya 2016

**PEMODELAN KURVA KARAKTERISTIK INVERSE NON-
STANDAR PADA RELE ARUS LEBIH DIGITAL DENGAN
METODE INTERPOLASI LAGRANGE**

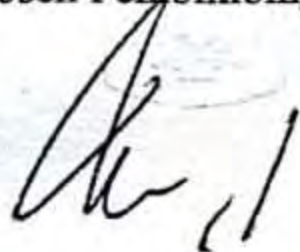
TUGAS AKHIR

**Diajukan untuk Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada**

**Bidang Studi Teknik Sistem Tenaga
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui:

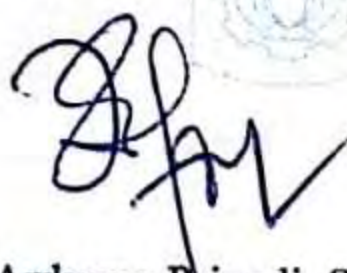
Dosen Pembimbing I



Dr. Ir. Margo Pujiانتara, MT.
NIP. 196603181990101001

Institut
Teknologi
Sepuluh Nopember

Dosen Pembimbing II



Dr. Eng. Ardyono Priyadi, ST., M. Eng.
NIP. 197309271998031004

Institut
Teknologi
Sepuluh Nopember



Pemodelan Kurva Karakteristik Inverse Non-Standar pada Rele Arus Lebih Digital dengan Metode Interpolasi Lagrange

Nama : Nurio Herlambang
Pembimbing I : Dr. Ir. Margo Pujiantara, M.T.
Pembimbing II : Dr. Ardyono Priyadi, S.T., M.Eng.

ABSTRAK

Saat ini, pengaman pada sistem tenaga listrik telah memakai rele digital sebagai peralatan pengaman. Rele arus lebih dengan kurva karakteristik *inverse* sangat bermanfaat untuk mengamankan gangguan akibat *overload*/beban lebih, karena bekerja dengan waktu tunda yang tergantung dari besarnya arus secara terbalik (*inverse time*), makin besar arus maka makin kecil waktu tundanya. Kurva karakteristik *inverse* rele arus lebih berdasarkan standar IEC seringkali dapat terjadi tumpang tindih dengan kurva lainnya selama proses koordinasi yang disebabkan oleh kebutuhan beban pada industri yang sangat kompleks. Hal ini dapat menyebabkan operasi *tripping* yang tidak akurat. Maka diperlukan kurva karakteristik *inverse* non-standar yang dapat menyesuaikan dengan kebutuhan beban. Untuk dapat menghasikan kurva yang non-standar pada rele arus lebih dibutuhkan sebuah pemodelan kurva dengan menggunakan metode Interpolasi Lagrange. *Personal Computer* (PC) digunakan sebagai media untuk mendesain kurva karakteristik yang diinginkan. Pemodelan kurva non-standar dilakukan di *software* MATLAB dengan membuat GUI dari program pemodelan kurva. GUI yang dirancang memiliki fungsi untuk mendesain kurva karakteristik *inverse* yang non-standar. Pada penerapan kurva non-standar pada ETAP 12.6, dengan nilai arus beban sebesar 657,7 A, waktu tripnya adalah 1,71 detik lebih cepat dibandingkan waktu trip pada kurva standar yaitu 2,13 detik. Hal ini dikarenakan pemodelan kurva non-standar tidak menggunakan persamaan lagi untuk menghitung nilai waktu trip, namun kurva didesain oleh user dengan cara membuat titik-titik data baru. Hasil pengujian prototipe rele arus lebih digital menunjukkan hasil yang mendekati dengan data target pada ETAP dengan rata-rata error 1,753 %.

Kata kunci : Interpolasi Lagrange, Rele Arus Lebih, Kurva Karakteristik Inverse,

Halaman ini sengaja di kosongkan

Non-Standard Inverse Characteristic Curve Modeling of Digital Overcurrent Relay with Lagrange Interpolation Methode

Name : Nurio Herlambang
Advisor I : Dr. Ir. Margo Pujiantara, M.T.
Advisor II : Dr. Ardyono Priyadi, S.T., M.Eng.

ABSTRACT

Currently, protection of electrical power system has used digital relay as a protection device. Overcurrent relay with inverse characteristic is very useful for protecting from overload, because it works with delay time which depends on current in inverse time, the greater the current the so delay time getting faster. Inverse characteristic curve of overcurrent relay based on IEC standard sometimes can overlap with one another during the coordination process because load demand in industrial is very complex. This problem can cause erroneous tripping operation. So it is needed non-standar inverse characteristic curve which can adjust to load demand. To obtain non-standard curve on overcurrent relay, it is needed a curve modeling using lagrange interpolation. PC is used as a device to design a desired characteristic curve. Modeling of non-standard curve is performed on MATLAB software with designing GUI of curve modeling program. Designed GUI has function to design non-standar inverse characteristic curve. When non-standard curve being tested on ETAP 12.6, load current 657,7 ampere, so delay time is 1,71 second faster than delay time in standar curve. Because modeling of non-standard curve doesn't require equation anymore, but the curve is designed by user by creating new data points. The testing result of digital overcurrent relay shows the result closer to target data of ETAP with mean percent error 1,753 %.

Keywords : Lagrange Interpolation, Overcurrent Relay, Inverse Characteristic Curve

Halaman ini sengaja di kosongkan

DAFTAR ISI

ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Permasalahan	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Metodologi	3
1.6 Sistematika Penulisan	4
1.7 Relevansi	5
BAB II TEORI PENUNJANG	7
2.1 Gangguan-gangguan Pada Sistem Tenaga Listrik	7
2.2 Sifat-sifat Gangguan	8
2.3 Rele Pengaman pada Sistem Tenaga Listrik	9
2.4 Rele Arus Lebih	10
2.4.1 Kurva Karakteristik Rele Arus Lebih	11
2.4.2 Rele Arus Lebih <i>Inverse</i>	11
2.5 Interpolasi Lagrange	14
2.6 Mikrokontroler	17
2.6.1 ARM	17
BAB III PERANCANGAN SOFTWARE DAN HARDWARE	19
3.1 Sistem Pengaman PT. HESS Indonesia	19
3.2 Pemodelan Kurva Karakteristik Inverse Non-standar	22
3.2.1 Algoritma Interpolasi Lagrange	22
3.3 Perancangan Hardware	24
3.4 Program Interpolasi Lagrange pada Rele Digital	25
BAB IV HASIL PENGUJIAN DAN ANALISIS	27
4.1 Pengujian Kurva Karakteristik Normal Inverse	29
4.2 Pengujian Keakuratan	30
4.2.1 Pengujian Keakuratan dengan 10 data input	31
4.2.2 Pengujian Keakuratan dengan 14 data input	32
4.2.3 Pengujian Keakuratan dengan 27 data input	34

4.3 Pengujian Kurva Karakteristik Inverse Non-standar	36
4.3.1 Penerapan Kurva Non-standar pada ETAP	39
4.3.2 Pengambilan data arus dan waktu Kurva Non-Standar	40
4.4 Pengujian Hardware	42
BAB V PENUTUP	47
5.1 Kesimpulan.....	47
5.2 Saran.....	48
DAFTAR PUSTAKA.....	49
LAMPIRAN	I
RIWAYAT HIDUP.....	XVII

DAFTAR GAMBAR

Gambar 2.1. Diagram konsep kerja rele pengaman	9
Gambar 2.2. Karakteristik Standard Inverse.....	12
Gambar 2.3. Karakteristik Very Inverse.....	12
Gambar 2.4. Karakteristik Extremely Inverse.	13
Gambar 2.5. Karakteristik Long Time Inverse.	13
Gambar 2.6. Mikrokontroler ARM.....	17
Gambar 3.1. Kurva inverse rele arus lebih PT. HESS Indonesia.....	19
Gambar 3.2. <i>Single line Diagram</i> PT. HESS Indonesia	21
Gambar 3.3. Flowchart Pemodelan Kurva	22
Gambar 3.4. Flowchart Interpolasi Lagrange	23
Gambar 3.5. Hardware Mikrokontroller ARM	24
Gambar 3.6. Rangkaian LCD grafik 128x64	24
Gambar 3.7. <i>Prototype</i> Rele Digital	25
Gambar 3.8. Flowchart Program Lagrange di Mikrokontroler	25
Gambar 4.1. GUI Pemodelan Kurva dengan Interpolasi Lagrange	27
Gambar 4.2. Hasil simulasi pemodelan kurva inverse standar	30
Gambar 4.3. Kurva hasil simulasi pengujian dengan 10 data input.....	32
Gambar 4.4. Kurva hasil simulasi pengujian dengan 14 data input.....	33
Gambar 4.5. Kurva hasil simulasi pengujian dengan 27 data input.....	35
Gambar 4.6. Perbandingan Error pengujian	35
Gambar 4.7. Kurva inverse pada rele arus lebih PT. HESS Indonesia	36
Gambar 4.8. Simulasi pemodelan kurva inverse non-standar.....	37
Gambar 4.9. Hasil simulasi pemodelan kurva inverse non-standar	38
Gambar 4.10. Kurva inverse standar pada ETAP	39
Gambar 4.11. Kurva inverse non-standar pada ETAP.....	40
Gambar 4.12. Pengambilan data arus uji dan trip waktu pada <i>Star System</i>	41
Gambar 4.13. Pengujian prototipe rele arus lebih digital	44
Gambar 4.14. Perbandingan Kurva Non-standar ETAP dengan Hardware.....	46

Halaman ini sengaja di kosongkan

DAFTAR TABEL

Tabel 3.1. Setting Rele Pengaman PT. HESS Indonesia	20
Tabel 4.1. Data Kurva Rele R-VCB-09	29
Tabel 4.2. Data Kurva Rele R-VCB-09 dengan 10 data input	31
Tabel 4.3. Hasil pengujian Rele R-VCB-09 dengan 10 data input	31
Tabel 4.4. Data Kurva Rele R-VCB-09 dengan 14 data input	32
Tabel 4.5. Hasil pengujian Rele R-VCB-09 dengan 14 data input	33
Tabel 4.6. Data Kurva Rele R-VCB-09 dengan 27 data input	34
Tabel 4.7. Hasil pengujian Rele R-VCB-09 dengan 27 data input	34
Tabel 4.8. Data Kurva Non-Standar Rele R-VCB-09	38
Tabel 4.9. Data arus uji dan waktu trip pada <i>Star System</i>	42
Tabel 4.10. Data arus uji dan waktu trip pada sisi sekunder	43
Tabel 4.11. Data Kurva Non-standar pada sisi sekunder.....	43
Tabel 4.12. Hasil Pengujian prototipe rele digital	45

Halaman ini sengaja di kosongkan

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Saat ini, pengaman pada sistem tenaga listrik telah memakai rele digital sebagai peralatan pengaman. Rele arus lebih dengan kurva karakteristik *inverse* sangat bermanfaat untuk mengamankan gangguan akibat *overload*/beban lebih, karena bekerja dengan waktu tunda yang tergantung dari besarnya arus secara terbalik (*inverse time*), makin besar arus maka makin kecil waktu tundanya. Kurva karakteristik *inverse* pada OCR sesuai dengan standar IEC ada 4 jenis, yaitu : *standard inverse* (SI), *very inverse* (VI), *extreme inverse* (EI), *long time inverse* (LTI) [1].

Kurva karakteristik *inverse* rele arus lebih berdasarkan standar IEC seringkali dapat terjadi tumpang tindih dengan kurva lainnya selama proses koordinasi yang disebabkan oleh kebutuhan beban pada industri yang sangat kompleks. Hal ini dapat menyebabkan operasi *tripping* yang tidak akurat. Maka diperlukan kurva karakteristik *inverse* non-standar yang dapat disesuaikan dengan kebutuhan beban. Untuk dapat menghasilkan kurva yang non-standar pada rele arus lebih dibutuhkan sebuah pemodelan kurva [2].

Pemodelan kurva karakteristik rele arus lebih awalnya dimulai dengan rele analog seperti rele elektromekanik dan rele statis. Rele digital menggunakan DSP (*Digital Signal Processor*) untuk proses proteksinya. DSP akan meningkatkan keandalan dan fleksibilitas dari pengaman karena DSP adalah perangkat berkecepatan tinggi. Pada rele digital, pemodelan kurva karakteristik menggunakan komputer untuk menggambar kurva dan menghitung *time delay* [3]. Karena beban pada industri sangat kompleks, hal ini dapat menyebabkan proses koordinasi menjadi tidak akurat. Pemodelan berdasarkan persamaan matematik biasa tidak cocok untuk menyelesaikan sistem dengan beban yang kompleks.

Fuzzy Logic (FL) dapat digunakan untuk menyelesaikan permasalahan yang kompleks seperti pengklasifikasian, pengolahan *database*, pengaturan otomatis, pemodelan, serta prediksi *time-series*. Neural Network digunakan untuk memodelkan kurva karakteristik rele arus lebih. Penggabungan FL dan Neural Network memungkinkan untuk

memodelkan kurva karakteristik standar rele arus lebih. Penggabungan FL dan Neural Network dikenal dengan ANFIS.

Pada tugas akhir akan dilakukan pemodelan kurva karakteristik *inverse* non-standar rele arus lebih dengan menggunakan metode Interpolasi Lagrange. Interpolasi adalah metode untuk menghasilkan titik-titik data baru dalam suatu jangkauan dari suatu data-data yang diketahui. Kemudian, akan dilakukan pengujian keakuratan kurva dengan jumlah *data learning* yang berbeda.

1.2 Permasalahan

Perumusan masalah pada tugas akhir ini adalah sebagai berikut :

1. Bagaimana mendapatkan titik-titik data baru untuk membentuk kurva *inverse* non-standar.
2. Memodelkan kurva non-standar pada PC.
3. Melakukan *learning* pada data kurva *inverse* non-standar.
4. Menguji hasil pemodelan dan *learning* metode Interpolasi Lagrange dengan beberapa arus uji yang dikehendaki.
5. Penerapan kurva *inverse* non-standar pada *software* ETAP 12.6

1.3 Batasan Masalah

Batasan masalah pada tugas akhir ini adalah sebagai berikut :

1. Kurva yang akan dimodelkan hanya kurva *inverse*.
2. *Software* yang digunakan untuk memodelkan kurva adalah MATLAB.
3. Pengujian dilakukan dengan menjalankan simulasi pemodelan kurva *inverse* pada GUI program pemodelan di MATLAB.
4. Hasil pengujian hanya ditinjau dari waktu trip pada pemodelan kurva *inverse* dengan beberapa arus uji yang dikehendaki *user*.

1.4 Tujuan

Tujuan pada tugas akhir ini adalah :

1. Memodelkan kurva karakteristik *inverse* non-standar dengan metode Interpolasi Lagrange.
2. Penerapan metode pemodelan kurva karakteristik *inverse* dengan menggunakan interpolasi lagrange

3. Mendapatkan kinerja rele arus lebih dengan tingkat keakuratan tinggi

1.5 Metodologi

Metodologi yang digunakan dalam penyusunan tugas akhir ini adalah sebagai berikut :

1. Studi Literatur
Mempelajari literatur mengenai teori penunjang tentang rele arus lebih dan definisi kurva *inverse* non-standar yang berasal dari *paper*, jurnal, dan *text book*. Selain itu mempelajari metode Interpolasi Lagrange dan cara membuat *interface* dari program pemodelan kurva *inverse* non-standar.
2. Pengumpulan Data
Diperlukannya pengumpulan data untuk mendukung pengujian simulasi program. Data-data yang diperlukan sebagai masukan dalam pemodelan kurva adalah arus beban dan waktu trip CB. Data-data ini diperoleh dari standar IEC untuk kurva karakteristik *inverse* rele arus lebih dan dari IEEE Transaction.
3. Pemodelan dan simulasi
Melakukan pembuatan program pemodelan kurva *inverse* menggunakan *software* MATLAB beserta *interface* dari program pemodelan kurva. Data yang diperoleh kemudian digunakan sebagai *input* pada program interpolasi lagrange. Pada bagian ini penulis juga akan melakukan pengambilan titik-titik data yang baru berupa arus gangguan dan waktu trip CB melalui GUI dari program pemodelan kurva. Data baru tersebut selanjutnya akan digabungkan dengan sebagian data kurva standar yang sudah ada dan kemudian dilakukan pengujian dengan beberapa arus uji yang dikehendaki.
4. Pengujian kurva non-standar
Pada tahap ini akan dilakukan pengujian terhadap pemodelan kurva non-standar dan membandingkannya dengan waktu trip dari *learning* interpolasi lagrange.
5. Analisis
Dilakukan analisis dari hasil pengujian berupa waktu trip yang akan didapatkan nilai error antara waktu trip dari data kurva non-standar dengan hasil *learning* interpolasi lagrange.

6. Kesimpulan

Langkah akhir dari penyusunan tugas akhir ini adalah pembuatan kesimpulan dari hasil analisis dan pengujian simulasi yang telah dilakukan. Selain itu, akan diberikan saran dan rekomendasi mengenai kelangsungan penelitian yang telah dilakukan pada pemodelan kurva karakteristik rele arus lebih dikemudian hari.

1.6 Sistematika Penulisan

Untuk mempermudah pembahasan yang akan dilakukan, tugas akhir ini terbagi menjadi lima bab dengan sistematika penulisan sebagai berikut :

BAB I : PENDAHULUAN

Bab ini membahas mengenai latar belakang, perumusan masalah, tujuan, metodologi pengerjaan tugas akhir, sistematika pembahasan dan relevansi dari tugas akhir ini.

BAB II : TEORI PENUNJANG

Pada bab ini akan dijelaskan teori-teori penunjang yang digunakan dalam pengerjaan tugas akhir ini.

BAB III : PERANCANGAN SOFTWARE SIMULASI

Bab ini akan menjelaskan perancangan sistem sebelum dilakukan simulasi berupa pemodelan kurva dan *flowchart* pemodelan kurva *inverse* non-standar.

BAB IV : HASIL PENGUJIAN DAN ANALISIS

Pada bab ini akan dibahas mengenai hasil simulasi yang telah dilakukan serta analisis terhadap penerapan pemodelan kurva karakteristik *inverse* non-standar dengan interpolasi lagrange. Pengujian dilakukan dengan jumlah data *learning* yang berbeda-beda.

BAB V : PENUTUP

Bab ini berisi kesimpulan dan saran dari hasil simulasi, studi literatur dan analisis yang telah dilakukan.

1.7 Relevansi

Tugas akhir ini diharapkan dapat memberikan manfaat, yaitu berkontribusi meningkatkan ilmu pengetahuan dan teknologi, khususnya tentang studi pemodelan kurva karakteristik rele arus lebih pada industri. Kemudian menjadi referensi dan rekomendasi kedepannya dapat melakukan pemodelan berbagai jenis kurva karakteristik *inverse* lainnya baik yang standar maupun non-standar.

Halaman ini sengaja di kosongkan

BAB II

TEORI PENUNJANG

Berdasarkan standar ANSI/IEEE 100-1992, gangguan pada sistem tenaga listrik didefinisikan sebagai satu kondisi yang disebabkan oleh kegagalan pada suatu perangkat, komponen, atau suatu elemen untuk bekerja sesuai dengan fungsi kerjanya. Suatu gangguan pada sistem tenaga listrik dapat dikatakan berbahaya karena mempunyai efek, antara lain :

1. Menyebabkan terjadinya kerusakan pada peralatan-peralatan sistem tenaga listrik yang dapat disebabkan karena kenaikan suhu atau kapasitas yang berlebihan.
2. Dapat membahayakan keselamatan pekerja.
3. Menyebabkan suatu sistem bekerja tidak stabil.

2.1 Gangguan-gangguan Pada Sistem Tenaga Listrik

Gangguan yang terjadi pada sistem tenaga listrik memiliki besaran dan jenis yang beragam. Gangguan pada sistem tenaga listrik merupakan keadaan tidak normal yang dapat menyebabkan kontinuitas pelayanan tenaga listrik terganggu. Secara umum klasifikasi gangguan pada sistem tenaga listrik disebabkan 2 faktor, yaitu [4] :

1. Gangguan berasal dari sistem
2. Gangguan berasal dari luar sistem

Penyebab gangguan yang berasal dari dalam sistem antara lain :

1. Tegangan dan arus abnormal
2. Pemasangan yang kurang baik
3. Kesalahan mekanis karena proses penuaan
4. Beban lebih
5. Kerusakan material seperti isolator pecah, kawat putus, atau kabel cacat isolasinya

Untuk penyebab gangguan yang berasal dari luar sistem antara lain :

1. Gangguan-gangguan mekanis karena pekerjaan galian saluran lain. Gangguan ini biasa terjadi pada sistem kelistrikan awah tanah
2. Pengaruh cuaca seperti hujan, angin, dan surja petir. Pada gangguan surja petir dapat mengakibatkan gangguan tegangan lebih dan dapat menyebabkan gangguan hubung singkat karena tembus isolasi peralatan (*breakdown*).
3. Pengaruh lingkungan seperti pohon tumbang, binatang dan benda-benda asing akibat kecerobohan manusia.

Dalam sistem tenaga listrik tiga fasa, gangguan-gangguan arus lebih yang mungkin dapat terjadi adalah sebagai berikut :

1. Gangguan beban lebih (*overload*)
Gangguan ini merupakan bukan gangguan murni, namun jika dibiarkan terus menerus terjadi maka dapat merusak peralatan listrik yang dialiri arus gangguan beban lebih tersebut. Ketika gangguan initerjadi, arus yang mengalir melebihi kapasitas peralatan listrik dan pengaman yang terpasang.
2. Gangguan hubung singkat (*shortcircuit*)
Gangguan hubung singkat dapat terjadi pada dua fasa, tiga fasa, satu fasa ke tanah, dua fasa ke tanah, atau tiga fasa ke tanah. Gangguan hubung singkat juga digolongkan menjadi dua yaitu, gangguan hubung singkat simetri dan gangguan hubung singkat tak simetri.

2.2 Sifat-sifat Gangguan

Jika ditinjau dari lamanya waktu gangguan terjadi, maka gangguan dapat dikelompokkan menjadi [5]:

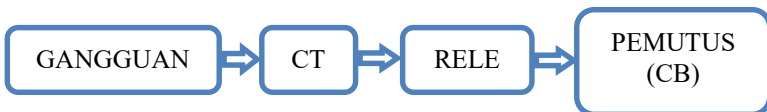
1. Gangguan yang bersifat temporer (sementara), yaitu gangguan yang dapat hilang dengan sendirinya atau dengan memutuskan atau mengisolir secara sesaat bagian yang mengalami gangguan dari sumber tegangannya. Gangguan sementara apabila tidak hilang dengan sendirinya atau diamankan dengan segera, maka dapat berubah menjadi gangguan permanen

2. Gangguan yang bersifat permanen, yaitu gangguan Diana untuk menghilangkan gangguan tersebut diperlukan tindakan perbaikan atau dengan menghilangkan penyebab gangguan tersebut.

Pada saat terjadi gangguan, maka akan mengalir arus yang besar pada asa yang mengalami gangguan menuju titik gangguan. Dimana arus gangguan yang terjadi memiliki nilai arus yang jauh lebih besar dibandingkan ranting arus maksimum yang diijinkan, sehingga dapat terjadi kenaikan suhu yang dapat menyebabkan kerusakan pada peralatan listrik.

2.3 Rele Pengaman pada Sistem Tenaga Listrik

Rele adalah suatu perlatan sistem tenaga listrik yang dirancang untuk memberikan sinyal kepada pemutus tenaga (CB), sehingga CB dapat memutuskan jaringan bila terjadi gangguan atau menghubungkan kembali penyaluran daya. Pemutusan ini dilakukan untuk memisahkan atau mengisolir bagian sistem tenaga listrik yang terjadi gangguan dengan yang tidak terkena gangguan.



Gambar 2.1. Diagram konsep kerja rele pengaman

Didalam rele terdapat 3 bagian, yaitu *sensing* elemen, *comparison* elemen dan *control* elemen setelah proses itu tercapai lalu dapat mengirimkan sinyal.

Terdapat syarat-syarat yang harus diperhatikan dalam penggunaan rele pengaman pada sistem proteksi, antara lain [6]:

- a. Selektifitas

Selektifitas pada sistem proteksi adalah kemampuan rele proteksi untuk melakukan pemilihan *tripping* secara tepat sesuai pada area yang terjadi gangguan. *Tripping* dengan menggunakan CB terdekat sesuai dengan rencana yang telah ditentukan saat mendesain sistem proteksi tersebut. Selektifitas sistem proteksi dapat dilakukan dengan dua cara yaitu waktu bertingkat (*time grading*) dan pembagian daerah pengamanan .

b. Sensitifitas

Sensitifitas disini adalah tingkat kepekaan suatu rele untuk dapat mendeteksi terhadap gangguan yang muncul. Suatu rele disebut sensitif apabila parameter operasi utamanya rendah. Artinya, semakin rendah besaran parameter penggerak maka perangkat tersebut dikatakan semakin sensitif.

Pada rele-rele numerik, sensitifitas tidak dikaitkan lagi pada perangkat kerasnya tetapi lebih pada aplikasi dan parameter peralatan pengukuran, seperti trafo arus (*current transformer*) atau trafo tegangan (*Voltage transformer*) yang dipakai.

c. Kecepatan

Kecepatan yang dimaksud adalah kecepatan untuk mengisolasi gangguan secepat dan sedini mungkin. Tujuan utamanya adalah mengamankan kontinuitas pasokan daya dengan cara mengisolasi gangguan.

d. Kehandalan

Rele dapat dikatakan handal dapat dihitung dari jumlah rele yang bekerja saat mengamankan gangguan berbanding dengan jumlah gangguan yang terjadi. Semakin tinggi kehandalan maka sistem tersebut semakin baik dan dapat meminimalkan terjadinya kerusakan akibat gangguan. Terdapat faktor-faktor yang mempengaruhi kehandalan, yaitu *dependable, secure, availability*.

2.4 Rele Arus Lebih

Rele arus lebih merupakan rele pengaman yang bekerja berdasarkan arus lebih yang disebabkan oleh terjadinya gangguan beban lebih dan gangguan hubung singkat yang kemudian akan memberikan perintah *trip* pada PMT sesuai dengan karakteristik waktunya. Apabila besarnya arus gangguan (I_f) melebihi suatu nilai tertentu (I_p) maka rele arus lebih akan bekerja. Dimana I_p adalah arus kerja rele berdasarkan sisi sekunder dari trafo arus (CT). Rele arus lebih akan bekerja jika memenuhi keadaan sebagai berikut :

$I_f > I_p \rightarrow$ rele bekerja (trip)

$I_f < I_p \rightarrow$ rele tidak bekerja

2.4.1 Kurva Karakteristik Rele Arus Lebih

Berdasarkan karakteristik waktunya, rele arus lebih dibedakan menjadi 4 jenis yaitu :

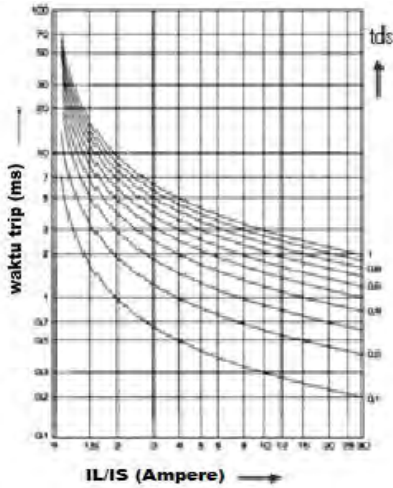
1. *Instantaneous Relay*
2. *Definite Relay*
3. *Inverse Relay*
4. *Inverse Definite Minimum Relay*

2.4.2 Rele Arus Lebih *Inverse*

Rele arus lebih dengan kurva karakteristik *inverse* sangat bermanfaat untuk mengamankan gangguan akibat *overload*/beban lebih, karena bekerja dengan waktu tunda yang berbanding terbalik dengan besarnya arus gangguan (*inverse time*), semakin besar arus gangguan maka semakin kecil waktu tundanya, sehingga semakin cepat waktu yang dibutuhkan untuk membuka PMT.

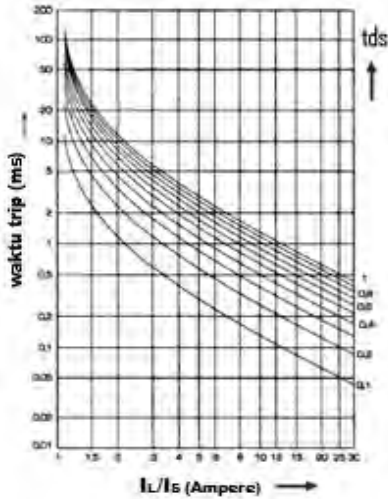
TCC (*Time-Current Characteristic*) adalah kurva waktu dan arus dengan skala dalam *time dial*. Jika semakin besar *time dial*, maka akan semakin lama waktu operasi dari rele arus lebih tersebut. Karakteristik kurva *inverse* ini dijelaskan dalam standar IEC 60255-3 dan IEEE 242-2001[7]. Kurva karakteristik *inverse* sering dijumpai dengan kurva *Inverse Definite Minimum Time (IDMT)*. Kurva IDMT merupakan kurva gabungan antara kurva *inverse* dan *definite*, yang artinya seiring dengan kenaikan arus yang semakin besar maka waktu operasi akan turun semakin cepat mendekati waktu *definite* minimumnya. Berdasarkan standar IEC, karakteristik *inverse* rele arus lebih dibagi menjadi 4 jenis, yaitu :

1. Karakteristik *standard inverse* (SI)
2. Karakteristik *very inverse* (SI)
3. Karakteristik *extremely inverse* (EI)
4. Karakteristik *long time inverse* (LTI)



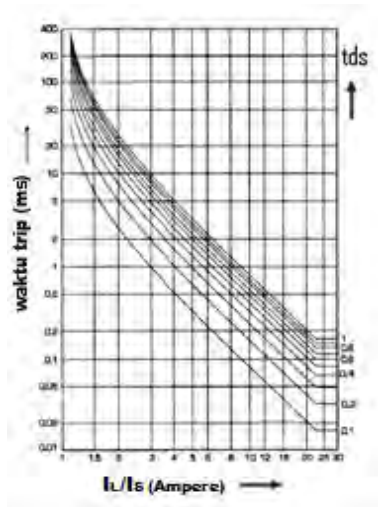
$$tcb = \frac{0,14 \times D}{\left(\frac{IL}{IS}\right)^{0,02} - 1} \quad (2.1)$$

Gambar 2.2. Karakteristik Standard Inverse.



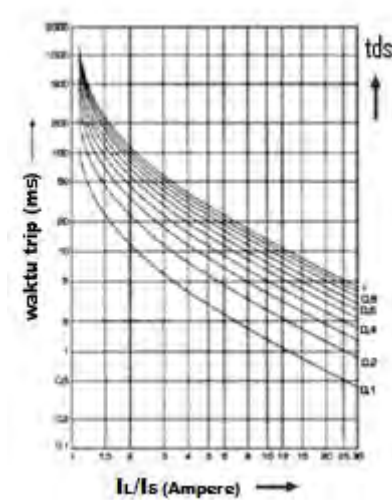
$$tcb = \frac{13,5 \times D}{\left(\frac{IL}{IS}\right) - 1} \quad (2.2)$$

Gambar 2.3. Karakteristik Very Inverse.



$$tcb = \frac{80 \times D}{\left(\frac{I_L}{I_S}\right)^2 - 1} \quad (2.3)$$

Gambar 2.4. Karakteristik Extremely Inverse.



$$tcb = \frac{120 \times D}{\left(\frac{I_L}{I_S}\right) - 1} \quad (2.4)$$

Gambar 2.5. Karakteristik Long Time Inverse.

2.5 Interpolasi Lagrange

Interpolasi adalah metode untuk menghasilkan titik-titik data baru dalam suatu jangkauan dari suatu data-data yang diketahui[8]. Interpolasi Lagrange diterapkan untuk mendapatkan fungsi polinomial $f(x)$ berderajat tertentu yang melewati sejumlah n titik data $f_1(x_1, y_1)$, $f_2(x_2, y_2)$, $f_3(x_3, y_3)$, ..., $f_n(x_n, y_n)$ dengan menggunakan pendekatan fungsi polinomial yang disusun dalam kombinasi deret dan didefinisikan dengan :

$$y = \sum_{i=1}^N y_i \prod_{j \neq i} \frac{(x - x_j)}{(x_i - x_j)} \quad (2.5)$$

Jumlah orde pada interpolasi lagrange adalah $n-1$. Bentuk umum interpolasi polinomial Lagrange order $n-1$ adalah:

$$f_n(x) = \sum_{i=0}^n L_i(x) f(x_i) \quad (2.6)$$

dengan

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j} \quad (2.7)$$

Simbol Π merupakan perkalian.

Dengan menggunakan persamaan (2.6) dan persamaan (2.7) dapat dihitung interpolasi Lagrange order yang lebih tinggi, misalnya untuk interpolasi Lagrange order 1, persamaan tersebut adalah:

$$f_1(x) = \sum_{i=0}^1 L_i(x) f(x_i) = L_0(x) f(x_0) + L_1(x) f(x_1) \quad (2.8)$$

$$L_0(x) = \left(\frac{x - x_1}{x_0 - x_1} \right) \quad (2.9)$$

$$L_1(x) = \left(\frac{x - x_0}{x_1 - x_0} \right) \quad (2.10)$$

Sehingga bentuk interpolasi polinomial Lagrange order 1 adalah:

$$f_1(x) = \left(\frac{x - x_1}{x_0 - x_1} \right) f(x_0) + \left(\frac{x - x_0}{x_1 - x_0} \right) f(x_1) \quad (2.11)$$

Dengan menggunakan persamaan (2.6) dan persamaan (2.7) dapat dihitung pula interpolasi Lagrange order 2 adalah:

$$F_2(x) = \sum_{i=0}^2 L_i(x) f(x_i) = L_0(x) f(x_0) + L_1(x) f(x_1) + L_2(x) f(x_2) \quad (2.12)$$

$$I=0 \quad L_0(x) = \left(\frac{x - x_1}{x_0 - x_1} \right) \left(\frac{x - x_2}{x_0 - x_2} \right) \quad (2.13)$$

$$I=1 \quad L_1(x) = \left(\frac{x - x_0}{x_1 - x_0} \right) \left(\frac{x - x_2}{x_1 - x_2} \right) \quad (2.14)$$

$$I=2 \quad L_2(x) = \left(\frac{x - x_0}{x_2 - x_0} \right) \left(\frac{x - x_1}{x_2 - x_1} \right) \quad (2.15)$$

Sehingga bentuk interpolasi polinomial Lagrange order 2 adalah:

$$f_2(x) = \left(\frac{x - x_1}{x_0 - x_1} \right) \left(\frac{x - x_2}{x_0 - x_2} \right) f(x_0) + \left(\frac{x - x_0}{x_1 - x_0} \right) \left(\frac{x - x_2}{x_1 - x_2} \right) f(x_1)$$

$$+ \left(\frac{x - x_0}{x_2 - x_0} \right) \left(\frac{x - x_1}{x_2 - x_1} \right) f(x_2) \quad (2.16)$$

Dengan menggunakan persamaan (2.6) dan persamaan (2.7) dapat dihitung pula interpolasi Lagrange order yang lebih tinggi, misalnya untuk interpolasi Lagrange order 3, persamaan tersebut adalah:

$$f_3(x) = \sum_{i=0}^3 L_i(x) f(x_i) = L_0(x) f(x_0) + L_1(x) f(x_1) + L_2(x) f(x_2) + L_3(x) f(x_3) \quad (2.17)$$

$$L_0(x) = \left(\frac{x - x_1}{x_0 - x_1} \right) \left(\frac{x - x_2}{x_0 - x_2} \right) \left(\frac{x - x_3}{x_0 - x_3} \right) \quad (2.18)$$

$$L_1(x) = \left(\frac{x - x_0}{x_1 - x_0} \right) \left(\frac{x - x_2}{x_1 - x_2} \right) \left(\frac{x - x_3}{x_1 - x_3} \right) \quad (2.19)$$

$$L_2(x) = \left(\frac{x - x_0}{x_2 - x_0} \right) \left(\frac{x - x_1}{x_2 - x_1} \right) \left(\frac{x - x_3}{x_2 - x_3} \right) \quad (2.20)$$

$$L_3(x) = \left(\frac{x - x_0}{x_3 - x_0} \right) \left(\frac{x - x_1}{x_3 - x_1} \right) \left(\frac{x - x_2}{x_3 - x_2} \right) \quad (2.21)$$

Sehingga bentuk interpolasi polinomial Lagrange order 3 adalah:

$$\begin{aligned} f_3(x) = & \left(\frac{x - x_1}{x_0 - x_1} \right) \left(\frac{x - x_2}{x_0 - x_2} \right) \left(\frac{x - x_3}{x_0 - x_3} \right) f(x_0) + \left(\frac{x - x_0}{x_1 - x_0} \right) \left(\frac{x - x_2}{x_1 - x_2} \right) \left(\frac{x - x_3}{x_1 - x_3} \right) f(x_1) \\ & + \left(\frac{x - x_0}{x_2 - x_0} \right) \left(\frac{x - x_1}{x_2 - x_1} \right) \left(\frac{x - x_3}{x_2 - x_3} \right) f(x_2) + \left(\frac{x - x_0}{x_3 - x_0} \right) \left(\frac{x - x_1}{x_3 - x_1} \right) \left(\frac{x - x_2}{x_3 - x_2} \right) f(x_3) \end{aligned}$$

2.6 Mikrokontroler

Mikrokontroller adalah sebuah rangkaian digital diintegrasi, yang merupakan jantung dari setiap aplikasi kontrol modern. Mikroprosesor dan mikrokontroler sama, tetapi sejauh domain aplikasi yang bersangkutan, arsitektur mereka berbeda. Mikroprosesor atau CPU utamanya digunakan untuk aplikasi komputasi kecepatan tinggi. Dilain pihak mikrokontroller digunakan dalam automasi dan aplikasi control. Namun sekarang domain aplikasi dari mikrokontroller telah diperluas ke sector komputasi. Contohnya banyak tertanam mikrokontroler pada modem, disk drives, floppy disk, dll. Sebelum kedatangan mikrokontroller, mikroprocessor telah dipilih untuk merealisasikan system control. Meskipun mikrokontroler telah dianggap menjadi pengganti mikroprocessor, dalam industri masih menggunakan mikroprocessor dikarenakan mudah menyesuaikan, fleksibel, harga murah, dan kemudahan menulis serta penanganan kode. Untuk kedepannya dalam memahami mikrokontroler , melakukan penelitian pada mikroprosesor adalah penting.

2.6.1 ARM

ARM adalah prosesor dengan arsitektur set instruksi 32-bit RISC (Reduced Instruction Set Computer) yang dikembangkan oleh ARM Holdings. ARM merupakan singkatan dari Advanced RISC Machine (sebelumnya lebih dikenal dengan kepanjangan Acorn RISC Machine)



Gambar 2.6. Mikrokontroler ARM

Halaman ini sengaja di kosongkan

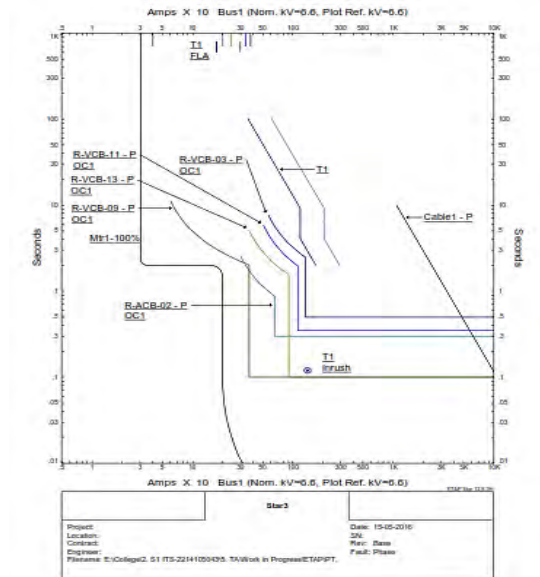
BAB III

PERANCANGAN SOFTWARE DAN HARDWARE

Pada bab ini akan dijelaskan perencanaan dan pembuatan software simulasi. Tugas akhir ini bertujuan untuk mendesain kurva karakteristik inverse non-standar dengan menggunakan metode interpolasi lagrange dengan menggunakan *software* MATLAB. Selain itu juga akan dilakukan pembuatan *Graphical User Interface* (GUI) dari program pemodelan kurva dengan metode interpolasi lagrange.

3.1 Sistem Pengaman PT. HESS Indonesia

Untuk melakukan pemodelan kurva *inverse*, dilakukan pengambilan data berupa data kurva *inverse* dari rele arus lebih. Pengambilan data dilakukan pada plan dari PT. HESS Indonesia menggunakan *software* ETAP 12.6. Gambar 3.1 merupakan grafik kurva *inverse* dari sistem kelistrikan dari PT. HESS Indonesia.

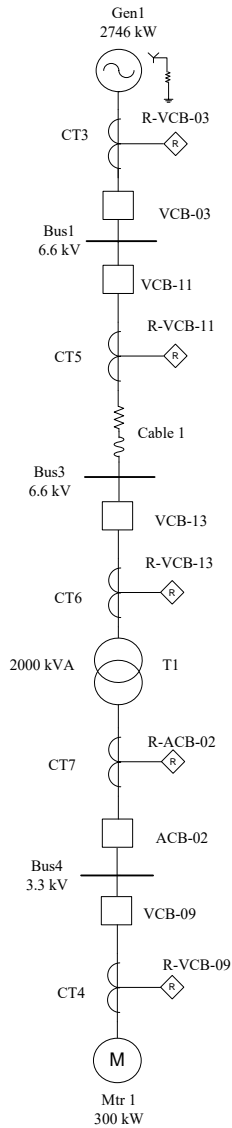


Gambar 3.1. Kurva inverse rele arus lebih PT. HESS Indonesia

Pada gambar 3.1 dapat dilihat bahwa terdapat tumpang tindih atau perpotongan antara kurva rele R-VCB-09 dan kurva rele R-ACB-02. Hal ini dapat membuat koordinasi proteksi menjadi tidak tepat. Untuk itu akan dilakukan pemodelan kurva standar menjadi non-standar pada kurva rele R-VCB-09. Tabel 3.1 menunjukkan *setting* rele arus lebih pada sistem kelistrikan PT. HESS Indonesia. *Single Line Diagram* dari PT HESS Indonesia ditunjukkan pada gambar 3.2.

Tabel 3.1. Setting Rele Pengaman PT. HESS Indonesia

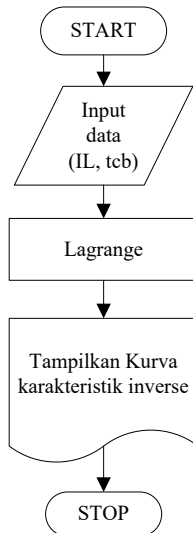
ID	CT Ratio	Setting	
		Curve	IEC- Normal Inverse
R-VCB-09	60/5	Current setting	1,35
		Time dial	0,65
		Current high set	12,35
		Time delay	0,1
		Curve	IEC- Normal Inverse
R-ACB-02	350/5	Current setting	1,15
		Time dial	0,15
		Current high set	3,75
		Time delay	0,3
		Curve	IEC- Normal Inverse
R-VCB-13	180/5	Current setting	1,35
		Time dial	0,3
		Current high set	5,05
		Time delay	0,,1
		Curve	IEC- Normal Inverse
R-VCB-11	400/5	Current setting	0,85
		Time dial	0,34
		Current high set	2,8
		Time delay	0,35
		Curve	IEC- Normal Inverse
R-VCB-03	400/1	Current setting	0,95
		Time dial	0,45
		Current high set	3,36
		Time delay	0,5
		Curve	IEC- Normal Inverse



Gambar 3.2. *Single line Diagram* PT. HESS Indonesia

3.2 Pemodelan Kurva Karakteristik Inverse Non-standar

Pada pemodelan kurva karakteristik inverse pada OCR data input yang digunakan yaitu data arus beban (IL) dan data waktu trip. Hasil dari training data akan berupa kurva karakteristik. Flowchart dari pemodelan kurva karakteristik inverse dengan interpolasi lagrange adalah sebagai berikut :



Gambar 3.3. Flowchart Pemodelan Kurva

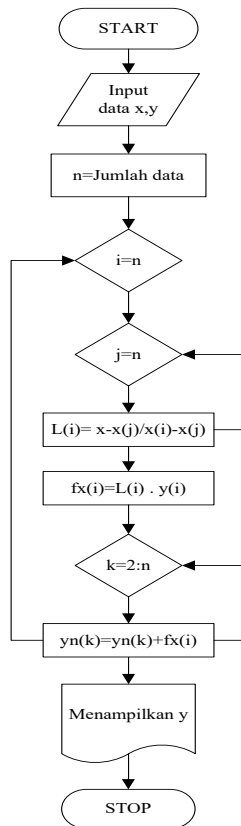
Flowchart diatas menjelaskan tentang proses pemodelan kurva karakteristik inverse. Dalam mendesain kurva karakteristik inverse tersebut akan digunakan software Matlab. Langkah awal dalam melakukan pemodelan kurva yaitu *user* menginputkan data training berupa data arus beban IL dan waktu trip. Semakin banyak jumlah data maka semakin kecil nilai errornya.

3.2.1 Algoritma Interpolasi Lagrange

Langkah pertama dalam interpolasi lagrange adalah menetapkan jumlah titik yang diketahui. Untuk memasukkan titik yang diketahui

dapat menggunakan dua array x dan y dengan jumlah data = jumlah titiknya. Dengan x merupakan nilai arus IL dan y adalah waktu trip.

Selanjutnya mencari nilai $L_i(x)$ sehingga diperlukan perulangan sebanyak jumlah titik yang diinputkan. $L_i(x)$ didapatkan dengan cara $Q_i(x)$ dibagi dengan $Q_i(x_i)$. $Q_i(x)$ merupakan hasil perkalian $(x-x_i)$, dengan x_i merupakan data yang diketahui, dan diperlukan perulangan sebanyak jumlah titik yang diinputkan untuk mencarinya. Setelah itu maka dapat dicari $L_i(x)$. Selanjutnya langkah terakhir mencari nilai y dengan mengalikan $L_i(x)$ dengan y_i .



Gambar 3.4. Flowchart Interpolasi Lagrange

3.3 Perancangan Hardware

Perancangan hardware rele digital menggunakan mikrokontroller ARM sebagai *prototype*-nya. Selain rele dikendalikan dengan mikrokontroller, rele akan menampilkan arus IL dan waktu trip dengan tampilan LCD grafik 128 x 64. Berikut adalah tampilan mikrokontroler ARM yang akan digunakan sebagai *prototype* rele digital.



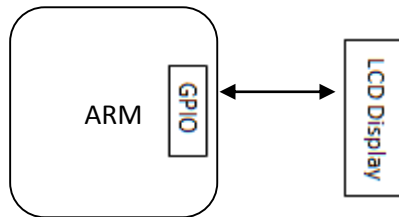
Gambar 3.5. Hardware Mikrokontroller ARM

LCD grafik 128 x 64 digunakan untuk menampilkan data rele dan untuk menunjukkan apakah rele sudah bekerja sesuai dengan besarnya arus gangguan yang terjadi. Berikut pada gambar 3.6 adalah rangkaian LCD grafik 128 x 64.



Gambar 3.6. Rangkaian LCD grafik 128x64

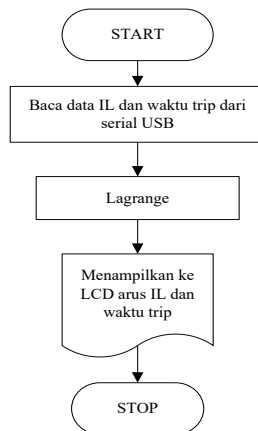
Gambar 3.7 menunjukkan *prototype* rele digital secara keseluruhan dengan LCD grafik terhubung ke mikrokontroler ARM melalui port *General-purpose input/output* (GPIO). GPIO adalah pin generik pada sirkuit terpadu (chip) yang perilakunya (termasuk apakah pin itu input atau output) dapat dikontrol (diprogram) oleh pengguna saat berjalan.



Gambar 3.7. *Prototype* Rele Digital

3.4 Program Interpolasi Lagrange pada Rele Digital

Input yang diperlukan pada rele digital ini adalah arus beban IL. Input dan output dari rele digital ini akan dikirim dan diterima menggunakan USB serial. Flowchart dari program interpolasi lagrange pada rele digital adalah sebagai berikut :

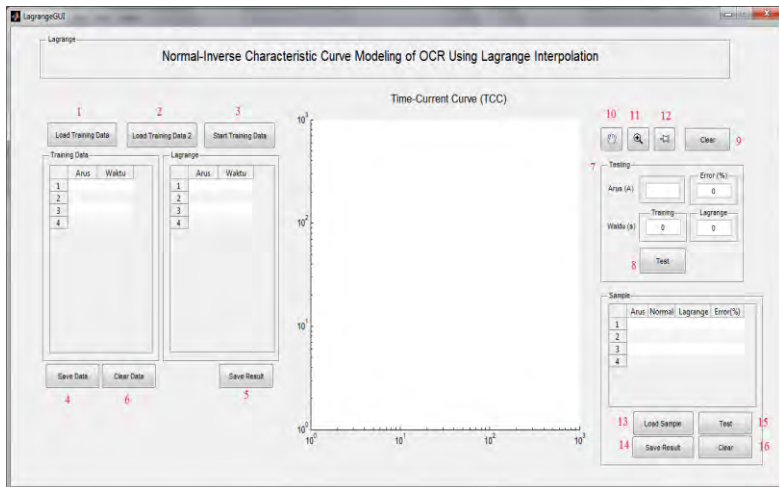


Gambar 3.8. Flowchart Program Lagrange di Mikrokontroler

Halaman ini sengaja di kosongkan

BAB IV HASIL PENGUJIAN DAN ANALISIS

Pada bab IV ini akan dilakukan pengujian dan analisis yang meliputi pengujian simulasi program yang telah dibuat dengan menggunakan metode interpolasi lagrange pada *software* MatLab.



Gambar 4.1. GUI Pemodelan Kurva dengan Interpolasi Lagrange

Keterangan dan fungsi pada gambar 4.1 :

1. **Load Training Data** merupakan tombol yang digunakan untuk mengisi/menginputkan data Training. Pada program ini format data yang digunakan dan dibaca adalah excel spreadsheet. Setelah memilih file excel, maka data akan ditampilkan pada tabel *Training Data* dan data input akan diplot pada grafik kurva.
2. **Load Training Data 2**, tombol untuk menampilkan atau memplot data kurva kedua pada grafik kurva.
3. **Start Training**, tombol untuk mengeksekusi pemodelan dengan interpolasi lagrange setelah data Training dimasukkan. Setelah

selesai training, hasil training akan dimunculkan pada tabel Lagrange dan diplot pada kurva TCC.

4. **Save Data**, tombol untuk menyimpan data training baru dalam format file excel spreadsheet jika ada perubahan data yang baru. Tombol ini digunakan jika ingin memodelkan kurva inverse non-standar dengan membuat titik-titik data baru.
5. **Save Result**, tombol untuk menyimpan data hasil pemodelan interpolasi lagrange dalam format file excel spreadsheet.
6. **Clear Data**, tombol untuk menghapus atau me-reset tabel *training data* dan tabel Lagrange menjadi 0 sebelum memulai atau mengulang training dengan data training yang lain.
7. Panel **Testing**, merupakan panel yang berisi fitur untuk menguji dan menampilkan hasil dari pengujian pemodelan interpolasi lagrange. Untuk mengujinya dengan memasukkan nilai arus di kolom *Arus (A)*.
8. **Test**, tombol untuk mengeksekusi pengujian hasil pemodelan interpolasi lagrange. Hasil pengujian berupa waktu trip lagrange yang akan tampil pada kolom waktu lagrange dan juga akan menampilkan waktu trip dari standar IEC sebagai pembandingnya dan nilai persen errornya.
9. **Clear**, tombol untuk menghapus atau me-reset kolom arus, waktu, dan juga eror menjadi nilai 0 dan juga me-reset ke kondisi *default* gambar kurva TCC.
10. **Hand**, tombol yang dapat digunakan untuk menggeser posisi objek/kurva pada grafik TCC.
11. **Zoom**, tombol yang digunakan untuk memperbesar dan memperkecil tampilan dari grafik TCC.
12. **Pin**, tombol yang digunakan untuk membuat titik-titik data baru pada kurva TCC dan menampilkan nilai arus dan waktu pada tabel *Training Data*. Fitur ini dapat digunakan untuk memodelkan kurva karakteristik inverse non-standar.
13. **Load Sample**, digunakan untuk menginputkan data uji jika memiliki beberapa *sample* data uji. Format data sample yang digunakan dan dibaca untuk pengujian adalah excel spreadsheet.
14. **Test**, tombol untuk mengeksekusi pengujian hasil pemodelan interpolasi lagrange. Hasil pengujian berupa waktu trip lagrange yang akan tampil pada tabel sample di kolom waktu

lagrange dan juga akan menampilkan waktu trip dari standar IEC sebagai pembandingnya dan nilai persen errornya.

15. **Save Result**, tombol untuk menyimpan data hasil pengujian interpolasi lagrange dengan beberapa data *sample* dalam format file excel spreadsheet.
16. **Clear Data**, tombol untuk menghapus atau me-reset semua data yang ada pada tabel *Sample* menjadi 0 sebelum memulai atau mengulang pengujian dengan data sample yang lain.

4.1 Pengujian Kurva Karakteristik Normal Inverse

Data pada pengujian *Standard Inverse* didapat dari rele OCR R-VCB-09 pada plan PT HESS Indonesia. Data rele OCR ditunjukkan pada tabel 4.1. Untuk kurva normal *inverse* perhitungannya menggunakan rumusan berikut:

$$tcb = \frac{0,14 \times D}{\left(\frac{I_L}{I_s}\right)^{0,02} - 1} \quad (4.1)$$

Tabel 4.1. Data Kurva Rele R-VCB-09

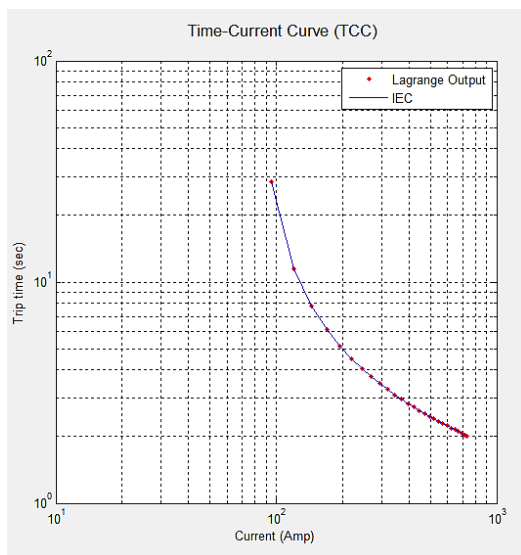
No	IL (A)	t(s)	No	IL (A)	t(s)
1	95	28,494	14	420	2,719
2	120	11,531	15	445	2,626
3	145	7,769	16	470	2,543
4	170	6,092	17	495	2,468
5	195	5,134	18	520	2,402
6	220	4,508	19	545	2,342
7	245	4,066	20	570	2,287
8	270	3,734	21	595	2,237
9	295	3,475	22	620	2,190
10	320	3,267	23	645	2,148
11	345	3,095	24	670	2,108
12	370	2,950	25	695	2,072
13	395	2,826	26	720	2,037
			27	735	2,018

Data pada tabel 4.1 didapatkan dari data *setting* rele R-VCB-09, dengan CT 60/5, tap *lowset* = 1.35, TD=0.65. Jadi $I_s=1.35 \times 60 = 81$ A. Sehingga waktu trip dapat diketahui.

$$t = \frac{0,14 \times 0,65}{(95/81)^{0,02} - 1}$$

t = 28,494 detik

Dari data pada tabel 4.1, maka didapatkan data kurva normal inverse dari hasil interpolasi lagrange yang sama dengan data kurva normal inverse IEC. Sehingga ketika data kurva hasil interpolasi lagrange diplot maka bentuk kurva akan sama dengan kurva normal inverse IEC seperti pada gambar 4.2.



Gambar 4.2. Hasil simulasi pemodelan kurva inverse standar

4.2 Pengujian Keakuratan

Pengujian keakuratan akan dilakukan dengan menguji kurva normal inverse dengan men-training hingga mendapat error terkecil. Pengujian ini dilakukan dengan merubah jumlah data input. Data rele yang akan di training berupa data input arus IL dimana IL merupakan arus beban. Selain data arus, akan di masukkan data tcb atau waktu trip CB sebagai target training.

4.2.1 Pengujian Keakuratan dengan 10 data input

Dapat dilihat pada tabel 4.2 terdapat 10 jumlah data dengan jarak nilai antar datanya adalah 75. Dari data pada tabel 4.2 akan dilakukan training dengan memasukkan beberapa arus uji sehingga didapat waktu trip CB hasil dari training interpolasi lagrange yang nantinya akan dibandingkan dengan hasil perhitungan waktu trip standar.

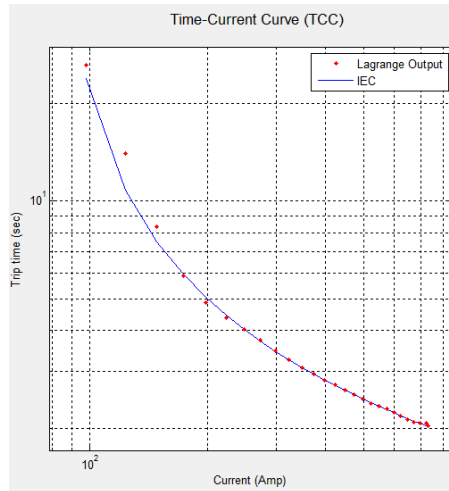
Tabel 4.2. Data Kurva Rele R-VCB-09 dengan 10 data input

No	IL (A)	t(s)	No	IL (A)	t(s)
1	95	28,494	6	470	2,543
2	170	6,092	7	545	2,342
3	245	4,066	8	620	2,190
4	320	3,267	9	695	2,072
5	395	2,826	10	735	2,018

Pada gambar 4.3, dapat dilihat titik merah merupakan kurva hasil interpolasi lagrange, sedangkan garis biru adalah kurva konvensional. Dari gambar, dapat dilihat bahwa dengan 10 jumlah data input, maka waktu trip hasil interpolasi lagrange belum mendekati target. Tabel 4.3 merupakan hasil pengujian dengan 10 data input, yang menunjukkan adanya nilai error pada saat pengujian.

Tabel 4.3. Hasil pengujian Rele R-VCB-09 dengan 10 data input

No	IL (A)	t(s)	Lagrange	Error	No	IL (A)	t(s)	Lagrange	Error
1	98	23,837	26,3376592	10,4918	14	423	2,707	2,71479842	0,2714
2	123	10,847	14,0574013	29,6015	15	448	2,615	2,62093947	0,2255
3	148	7,5032	8,35806747	11,3939	16	473	2,533	2,53228454	0,0357
4	173	5,9506	5,89731215	0,89533	17	498	2,46	24532908	0,2757
5	198	5,0452	4,86302563	3,61004	18	523	2,394	2,38802112	0,2617
6	223	4,4475	4,36515527	1,85133	19	548	2,335	2,33581814	0,0473
7	248	4,0209	4,02832957	0,18541	20	573	2,28	2,28998872	0,4181
8	273	3,6995	3,73367072	0,92446	21	598	2,231	2,2409056	0,4539
9	298	3,4476	3,46737213	0,57372	22	623	2,85	2,18304753	0,0937
10	323	3,2442	3,24195581	0,06843	23	648	2,143	2,12276162	0,9399
11	348	3,076	3,06361118	0,40154	24	673	2,104	2,0799421	1,1335
12	373	2,9342	2,92565133	0,2906	25	698	2,067	2,07237795	0,2407
13	398	2,8128	2,81390575	0,04015	26	723	2,033	2,06622856	1,6122
					27	732	2,022	2,03570357	0,6894



Gambar 4.3. Kurva hasil simulasi pengujian dengan 10 data input

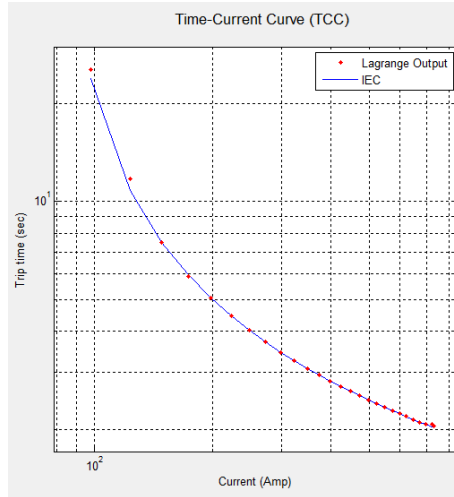
4.2.2 Pengujian Keakuratan dengan 14 data input

Untuk meningkatkan keakuratan maka jumlah data input perlu dinaikkan. Pada tabel 4.4 terdapat 14 jumlah data dengan jarak nilai antar datanya adalah 50.

Tabel 4.4. Data Kurva Rele R-VCB-09 dengan 14 data input

No	IL (A)	t(s)	No	IL (A)	t(s)
1	95	28,494	8	445	2,626
2	145	7,769	9	495	2,468
3	195	5,134	10	545	2,342
4	245	4,066	11	595	2,237
5	295	3,475	12	645	2,148
6	345	3,095	13	695	2,072
7	395	2,826	14	735	2,018

Pada gambar 4.4 menunjukan bahwa kurva interpolasi lagrange mulai mendekati kurva konvensional. Dengan meningkatkan jumlah data input menjadi 14 dengan jarak nilai antar data 50 maka menunjukkan perbaikan data output, dengan hasil mulai mendekati target.



Gambar 4.4. Kurva hasil simulasi pengujian dengan 14 data input

Tabel 4.5 merupakan hasil pengujian dengan 14 data input, yang menunjukkan adanya nilai error pada saat pengujian. Dengan semakin meningkatnya jumlah data input, maka nilai error menjadi menurun.

Tabel 4.5. Hasil pengujian Rele R-VCB-09 dengan 14 data input

No	IL (A)	t(s)	Lagrange	Error	No	IL (A)	t(s)	Lagrange	Error
1	98	23,84	25,49463	6,95515	14	423	2,707	2,70773859	0,010661
2	123	10,85	11,7355	8,19493	15	448	2,615	2,61498934	0,002045
3	148	7,503	7,468748	0,45866	16	473	2,533	2,53289425	0,011601
4	173	5,951	5,899775	0,85395	17	498	2,46	2,46013727	0,002578
5	198	5,045	5,048553	0,06729	18	523	2,394	2,39469467	0,016981
6	223	4,447	4,454848	0,16537	19	548	2,335	2,33461073	0,004401
7	248	4,021	4,02021	0,01651	20	573	2,28	2,27967785	0,034014
8	273	3,699	3,697619	0,05003	21	598	2,231	2,23101272	0,01043
9	298	3,448	3,447801	0,00604	22	623	2,185	2,18720191	0,096417
10	323	3,244	3,244884	0,02181	23	648	2,143	2,14213477	0,035843
11	348	3,076	3,075867	0,00311	24	673	2,104	2,09521654	0,407444
12	373	2,934	2,933793	0,01312	25	698	2,067	2,07125268	0,186286
13	398	2,813	2,812838	0,00217	26	723	2,033	2,0771146	2,147588
					27	732	2,022	2,04371678	1,085781

4.2.3 Pengujian Keakuratan dengan 27 data input

Untuk lebih meningkatkan keakuratan maka jumlah data input akan dinaikkan lagi. Pada tabel 4.6 terdapat 27 jumlah data dengan jarak nilai antar datanya diperkecil menjadi 25.

Tabel 4.6. Data Kurva Rele R-VCB-09 dengan 27 data input

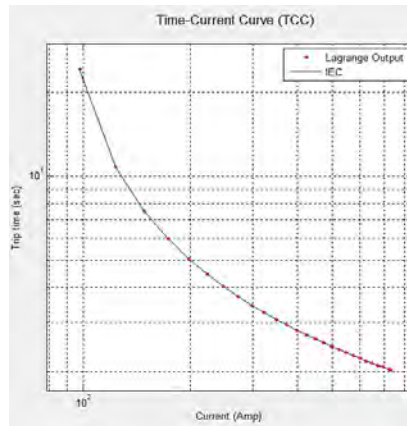
No	IL (A)	t(s)	No	IL (A)	t(s)
1	95	28,494	14	420	2,719
2	120	11,531	15	445	2,626
3	145	7,769	16	470	2,543
4	170	6,092	17	495	2,468
5	195	5,134	18	520	2,402
6	220	4,508	19	545	2,342
7	245	4,066	20	570	2,287
8	270	3,734	21	595	2,237
9	295	3,475	22	620	2,190
10	320	3,267	23	645	2,148
11	345	3,095	24	670	2,108
12	370	2,950	25	695	2,072
13	395	2,826	26	720	2,037
			27	735	2,018

Dan dengan semakin mendekatnya kurva interpolasi lagrange pada target menunjukkan bahwa nilai error menjadi semakin kecil, seperti yang terlihat pada tabel 4.7.

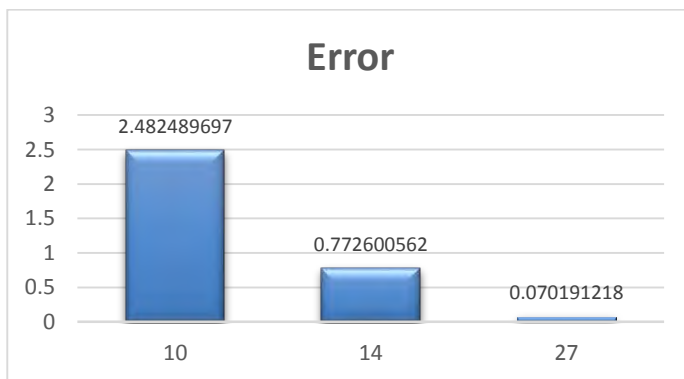
Tabel 4.7. Hasil pengujian Rele R-VCB-09 dengan 27 data input

No	IL (A)	t(s)	Lagrange	Error	No	IL (A)	t(s)	Lagrange	Error
1	98	23,837	24,244	1,710102723	14	423	2,707	2,707	1,11299E-07
2	123	10,847	10,839	0,06578905	15	448	2,615	2,615	1,17403E-07
3	148	7,503	7,504	0,005076529	16	473	2,533	2,533	1,4395E-07
4	173	5,951	5,951	0,000608604	17	498	2,460	2,460	2,05653E-07
5	198	5,045	5,045	0,000101557	18	523	2,394	2,394	3,43796E-07
6	223	4,447	4,447	2,21923E-05	19	548	2,335	2,335	6,7684E-07
7	248	4,021	4,021	6,11171E-06	20	573	2,280	2,280	1,8352E-06
8	273	3,699	3,699	2,06705E-06	21	598	2,231	2,231	4,5778E-06
9	298	3,448	3,448	8,43031E-07	22	623	2,185	2,185	1,3592E-05
10	323	3,244	3,244	4,09202E-07	23	648	2,143	2,143	6,62989E-05
11	348	3,076	3,076	2,34158E-07	24	673	2,104	2,104	0,000370029
12	373	2,934	2,934	1,56903E-07	25	698	2,067	2,067	0,002766171
13	398	2,813	2,813	1,22557E-07	26	723	2,033	2,033	0,02502293
					27	732	2,022	2,020	0,085203856

Dari gambar 4.5 didapatkan bentuk kurva interpolasi lagrange yang menindih persis kurva konvensional. Dengan meningkatkan jumlah data input menjadi 27 dengan jarak nilai antar data 25 maka diperoleh perbaikan data output.



Gambar 4.5. Kurva hasil simulasi pengujian dengan 27 data input

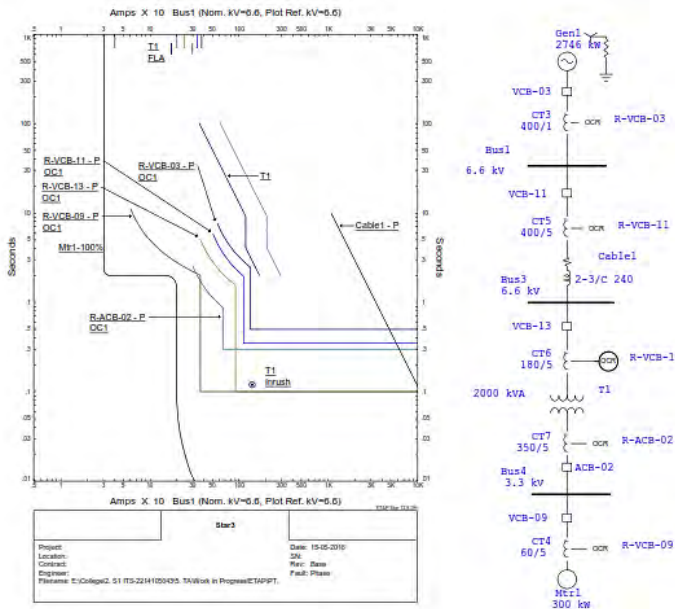


Gambar 4.6. Perbandingan Error pengujian

Pada gambar 4.6, menunjukkan perbandingan error pengujian dimana jumlah data 27 memiliki error terkecil yaitu 0,070191218 %.

4.3 Pengujian Kurva Karakteristik Inverse Non-standar

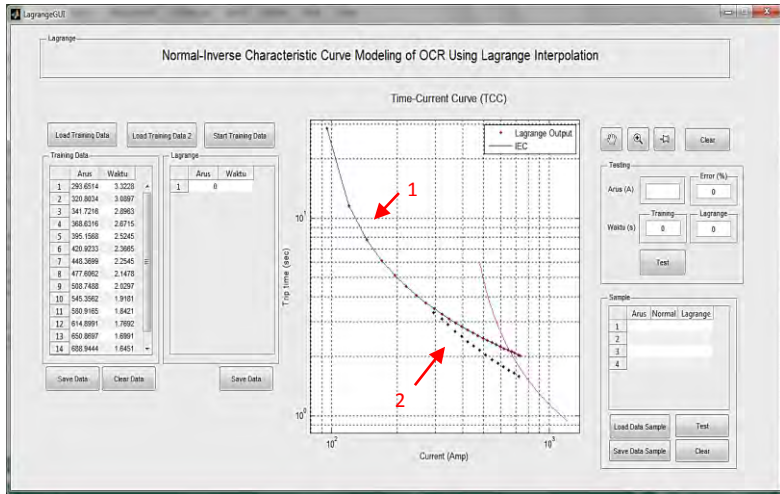
Gambar 4.7 merupakan kurva koordinasi proteksi dari PT. HESS Indonesia. Pada gambar tersebut menunjukkan adanya perpotongan atau tumpang tindih antara kurva rele R-VCB-09 dengan rele R-ACB-02. Hal ini dapat menyebabkan rele R-VCB-09 dan R-ACB-02 dapat bekerja secara bersama dalam waktu yang sama. Oleh sebab itu diperlukan pemodelan kurva inverse non-standar yang dapat menyelesaikan permasalahan tersebut.



Gambar 4.7. Kurva inverse pada rele arus lebih PT. HESS Indonesia

Konsep pemodelan kurva non-standar adalah user mendesain kurva dengan membuat titik-titik data baru yang digabungkan dengan data lama sehingga membentuk kurva yang tidak tumpang tindih dengan kurva lainnya, seperti yang ditunjukkan pada gambar 4.8 :

1. Poin 1 merupakan data input pada kurva standar yang digunakan sebagai data input untuk pemodelan kurva non-standar.
2. Poin 2 adalah titik-titik data baru yang telah dibuat oleh user yang selanjutnya akan digabungkan dengan data input lama pada poin 1 sehingga menjadi data input baru yang akan digunakan untuk pemodelan kurva non-standar.



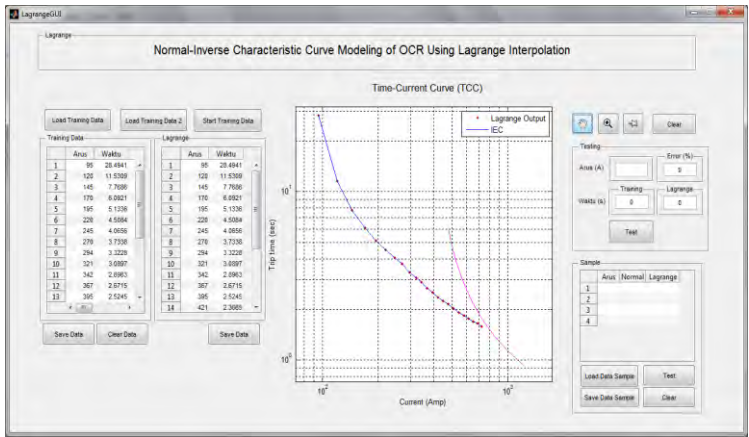
Gambar 4.8. Simulasi pemodelan kurva inverse non-standar

Tabel 4.8 adalah tabel yang berisi data kurva yang baru yang telah didesain oleh user yang akan digunakan sebagai data input untuk pemodelan kurva inverse non-standar.

Tabel 4.8. Data Kurva Non-Standar Rele R-VCB-09

No	IL (A)	t(s)	No	IL (A)	t(s)
1	95	28.494	14	420	2.365
2	120	11.531	15	445	2.259
3	145	7.769	16	470	2.175
4	170	6.092	17	495	2.088
5	195	5.134	18	520	2.015
6	220	4.508	19	545	1.964
7	245	4.066	20	570	1.896
8	270	3.637	21	595	1.843
9	295	3.251	22	645	1.734
10	320	3.02	23	670	1.686
11	345	2.833	24	720	1.611
12	370	2.672	25	735	1.588
13	395	2.527			

Gambar 4.9 adalah hasil simulasi pemodelan kurva dengan menggunakan data kurva yang baru. Dengan membuat membuat titik-titik data yang baru maka kurva non-standar dapat terbentuk, sehingga tidak terjadi perpotongan atau tumpang tindih antara kurva satu dengan yang lainnya.

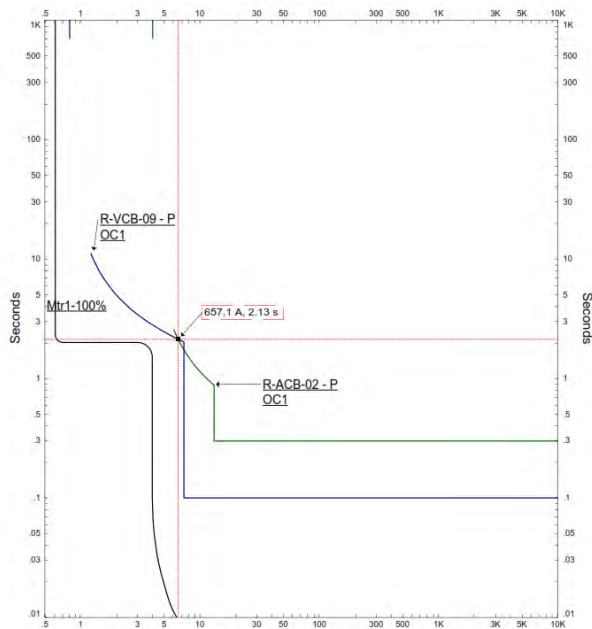


Gambar 4.9. Hasil simulasi pemodelan kurva inverse non-standar

4.3.1 Penerapan Kurva Non-standar pada ETAP

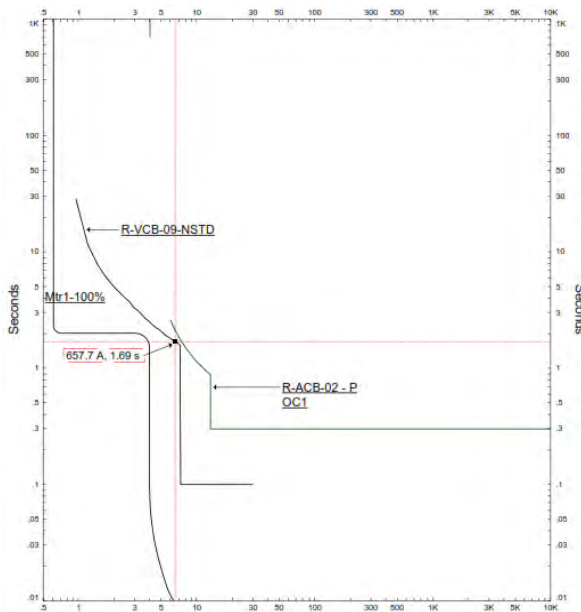
Setelah melakukan simulasi pemodelan kurva dengan menggunakan data kurva non-standar, maka langkah selanjutnya adalah menerapkan data kurva non-standar pada tabel 4.8 kedalam rele R-VCB-09 dengan menggunakan ETAP.

Pada gambar 4.10, terlihat bahwa kurva rele R-VCB-09 berpotongan dengan kurva rele R-ACB-02. Ketika terjadi arus beban lebih sebesar 657,7 A, maka rele R-VCB-09 dan R-ACB-02 akan merasakan gangguan dan memerintahkan CB untuk open, sehingga CB pada R-VCB-09 dan R-ACB-02 trip secara bersamaan pada waktu 2,13 detik. Hal ini menyebabkan koordinasi proteksi menjadi tidak tepat.



Gambar 4.10. Kurva inverse standar pada ETAP

Pada gambar 4.11 menunjukkan hasil penerapan dari data kurva non-standar pada rele R-VCB-09 sehingga terbentuk kurva non-standar.

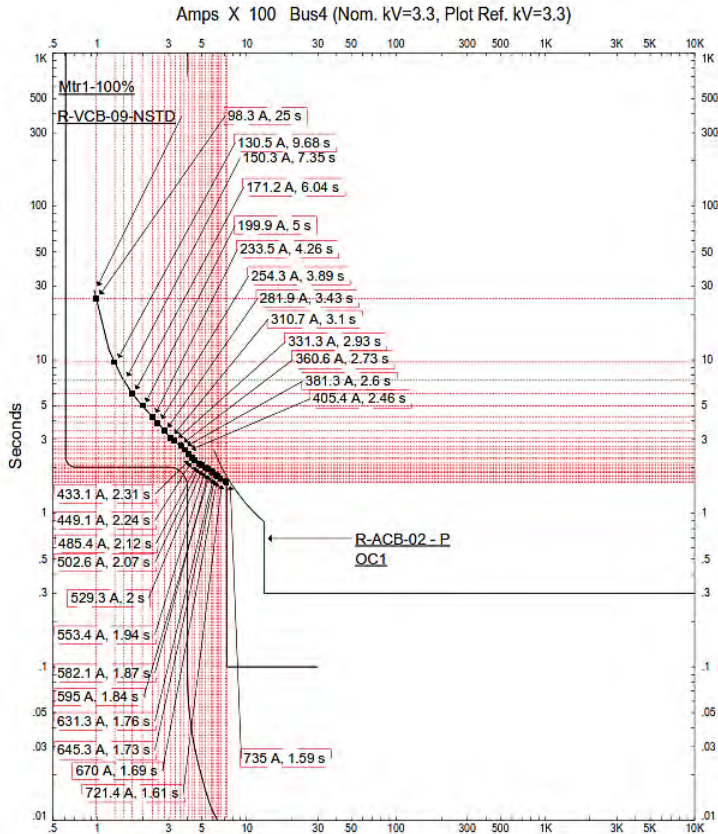


Gambar 4.11. Kurva inverse non-standar pada ETAP

Dengan mengubah kurva inverse menjadi non-standar, maka permasalahan perpotongan atau tumpang tindih kurva dapat diatasi. Dapat dilihat pada gambar 4.11, bahwa ketika terjadi arus beban lebih sebesar 657,7 A maka hanya rele R-VCB-09 saja yang dapat merasakan gangguan tersebut dan memerintahkan CB untuk open dengan waktu trip 1,71 detik.

4.3.2 Pengambilan data arus dan waktu Kurva Non-Standar

Dengan menggunakan kurva non-standar pada *Star System* ETAP yang ditunjukkan pada gambar 4.11, maka dapat dilakukan pengambilan data arus dan waktu yang akan digunakan sebagai data pembandingan ketika melakukan pengujian pada *hardware*. Gambar 4.12 merupakan data-data arus dan waktu yang diperoleh dari kurva non-standar pada *Star System* yang digunakan untuk pengujian prototipe rele arus lebih digital.



Gambar 4.12. Pengambilan data arus uji dan trip waktu pada *Star System*

Tabel 4.9 merupakan data-data arus dan waktu yang diperoleh dari kurva non-standar pada *Star System* yang akan digunakan untuk pengujian prototipe rele digital.

Tabel 4.9. Data arus uji dan waktu trip pada *Star System*

No	IL (A)	t(s)	No	IL (A)	t(s)
1	98.3	25	14	433.1	2.31
2	130.5	9.68	15	449.1	2.24
3	150.3	7.35	16	485.4	2.12
4	171.2	6.04	17	502.6	2.07
5	199.9	5.00	18	529.3	2.00
6	233.5	4.26	19	553.4	1.94
7	254.3	3.89	20	582.1	1.87
8	281.9	3.43	21	595	1.843
9	310.7	3.10	22	645	1.734
10	331.3	2.93	23	670	1.686
11	360.6	2.73	24	720	1.611
12	381.3	2.60	25	735	1.588
13	405.4	2.46			

4.4 Pengujian Hardware

Pada bagian ini, akan dilakukan pengujian kurva non-standar pada *hardware*, yaitu prototipe *Digital Protection Relay* berbasis mikrokontroler yang dilengkapi dengan LCD grafik. Pengujian dilakukan dengan menginjeksikan arus beban menggunakan *current injection* sehingga didapatkan waktu trip pada prototipe rele digital. Karena elemen *sensing* pada prototipe rele digital hanya dapat mendeteksi arus hanya sampai 10 A, maka arus beban yang diinjeksikan adalah arus beban yang melewati sisi sekunder CT dengan rasio CT yang digunakan 80:1.

Tabel 4.10 adalah data arus dan waktu pada tabel 4.9 yang telah dibagi dengan rasio CT 80:1 sehingga didapat arus beban yang melewati sisi sekunder CT.

Tabel 4.10. Data arus uji dan waktu trip pada sisi sekunder

No	IL (A)	t(s)	No	IL (A)	t(s)
1	1.23	25	14	5.41	2.31
2	1.63	9.68	15	5.61	2.24
3	1.88	7.35	16	6.07	2.12
4	2.14	6.04	17	6.28	2.07
5	2.5	5.00	18	6.62	2.00
6	2.92	4.26	19	6.92	1.94
7	3.18	3.89	20	7.28	1.87
8	3.52	3.43	21	7.44	1.843
9	3.88	3.10	22	8.06	1.734
10	4.14	2.93	23	8.38	1.686
11	4.51	2.73	24	9	1.611
12	4.77	2.60	25	9.19	1.588
13	5.07	2.46			

Langkah-langkah dalam pengujian prototipe rele digital

1. Mengubah data arus pada kurva non-standar pada tabel 4.8 menjadi besaran pada sisi sekunder CT

Tabel 4.11. Data Kurva Non-standar pada sisi sekunder

No	IL (A)	t(s)	No	IL (A)	t(s)
1	1.19	28.494	5	2.44	5.134
2	1.5	11.531	6	2.75	4.508
3	1.81	7.769	7	3.06	4.066
4	2.13	6.092	8	3.38	3.637

Lanjutan Tabel 4.11. Data Kurva Non-standar pada sisi sekunder

No	IL (A)	t(s)	No	IL (A)	t(s)
9	3.69	3.251	18	6.5	2.015
10	4	3.02	19	6.81	1.964
11	4.31	2.833	20	7.13	1.896
12	4.63	2.672	21	7.44	1.843
13	4.94	2.527	22	8.06	1.734
14	5.25	2.365	23	8.38	1.686
15	5.56	2.259	24	9	1.611
16	5.88	2.175	25	9.19	1.588
17	6.19	2.088			

2. Mendownload program interpolasi lagrange beserta data kurva non-standar pada tabel 4.11 kedalam prototipe rele digital.
3. Melakukan pengujian prototipe rele digital seperti berikut:

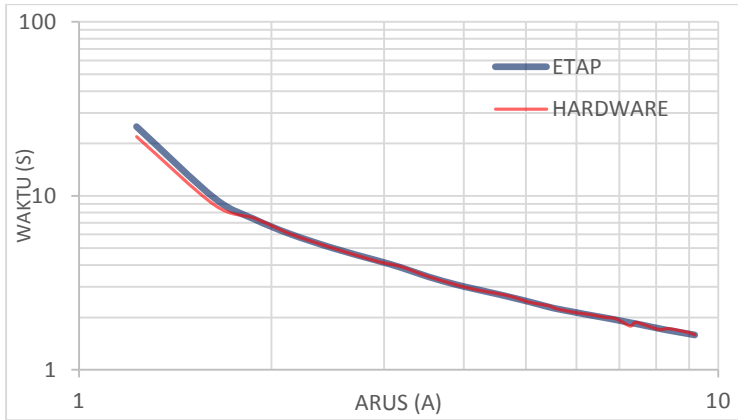


Gambar 4.13. Pengujian prototipe rele arus lebih digital

Pada gambar 4.13, merupakan langkah-langkah melakukan pengujian prototipe rele arus lebih digital. Pengujian dilakukan dengan menginjeksikan arus beban yang keluar dari *current injector* ke prototipe rele arus lebih digital berbasis mikrokontroler ARM. Pada gambar diatas, ketika arus beban yang dikeluarkan sebesar 5.04 A, maka prototipe rele arus lebih digital akan merasakan arus beban tersebut yang terbaca sebesar 5,075 A dan akan memerintahkan CB untuk trip dengan waktu trip 2,455 detik. Dengan menggunakan oscilloscope dapat dilihat panjang atau lebar gelombang ketika rele merasakan gangguan hingga memerintahkan CB untuk trip yang sebesar 2,445 detik.

Tabel 4.12. Hasil Pengujian prototipe rele digital

No	IL (A)	ETAP (s)	HARDWARE (s)	Error	No	IL (A)	ETAP (s)	HARDWARE (s)	Error
1	1.23	25	21.885	12.460	14	5.41	2.31	2.34	1.299
2	1.63	9.68	8.835	8.729	15	5.61	2.24	2.235	0.223
3	1.88	7.35	7.44	1.224	16	6.07	2.12	2.115	0.236
4	2.14	6.04	6.045	0.083	17	6.28	2.07	2.085	0.725
5	2.5	5	4.995	0.100	18	6.62	2	2.01	0.500
6	2.92	4.26	4.215	1.056	19	6.92	1.94	1.965	1.289
7	3.18	3.89	3.93	1.028	20	7.28	1.87	1.785	4.545
8	3.52	3.43	3.435	0.146	21	7.44	1.843	1.875	1.736
9	3.88	3.1	3.09	0.323	22	8.06	1.734	1.71	1.384
10	4.14	2.93	2.91	0.683	23	8.38	1.686	1.725	2.313
11	4.51	2.73	2.745	0.549	24	9	1.611	1.635	1.490
12	4.77	2.6	2.625	0.962	25	9.19	1.588	1.59	0.126
13	5.07	2.46	2.445	0.610					



Gambar 4.14. Perbandingan Kurva Non-standar ETAP dengan Hardware

Dari data pada tabel 4.12 dan gambar 4.14 diatas menunjukkan bahwa kurva hasil pengujian pada prototipe rele arus lebih digital dapat mengikuti kurva non-standar hasil penerapan di ETAP. Namun masih memiliki nilai error walaupun sangat kecil dan tidak terlalu jauh, dengan rata-rata error = 1.753 %.

Dari pengujian yang dilakukan dapat diketahui bahwa pemodelan kurva karakteristik *inverse* non-standar dapat dilakukan dengan menggunakan metode Interpolasi Lagrange. Dimana metode ini melakukan training data kurva non-standar dengan membuat titik-titik data baru sehingga kurva tidak berpotongan dengan kurva *inverse* pada rele arus lebih lainnya. Karena pemodelan kurva non-standar tidak ditentukan oleh persamaan lagi untuk menghitung waktu trip, tetapi kurva didesain oleh user.

LAMPIRAN

List Coding Program GUI

```
function varargout = LagrangeGUI(varargin)
% LAGRANGEGUI MATLAB code for LagrangeGUI.fig
%   LAGRANGEGUI, by itself, creates a new LAGRANGEGUI or
%   raises the existing
%   singleton*.
%
%   H = LAGRANGEGUI returns the handle to a new
%   LAGRANGEGUI or the handle to
%   the existing singleton*.
%
%   LAGRANGEGUI('CALLBACK',hObject,eventData,handles,...)
%   calls the local
%   function named CALLBACK in LAGRANGEGUI.M with the
%   given input arguments.
%
%   LAGRANGEGUI('Property','Value',...) creates a new
%   LAGRANGEGUI or raises the
%   existing singleton*. Starting from the left,
%   property value pairs are
%   applied to the GUI before LagrangeGUI_OpeningFcn gets
%   called. An
%   unrecognized property name or invalid value makes
%   property application
%   stop. All inputs are passed to
%   LagrangeGUI_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI
%   allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
LagrangeGUI

% Last Modified by GUIDE v2.5 15-May-2016 07:07:10

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
```

```

        'gui_OpeningFcn',
@LagrangeGUI_OpeningFcn, ...
        'gui_OutputFcn', @LagrangeGUI_OutputFcn,
...
        'gui_LayoutFcn', [] , ...
        'gui_Callback', []];
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before LagrangeGUI is made visible.
function LagrangeGUI_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)
% varargin   command line arguments to LagrangeGUI (see
VARARGIN)

% Choose default command line output for LagrangeGUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes LagrangeGUI wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command
line.
function varargout = LagrangeGUI_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see
VARARGOUT);

```

```

% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in PB_LOAD.
function PB_LOAD_Callback(hObject, eventdata, handles)
global myData;
global x;
global y;

[filename pathname]=uigetfile({'*.xlsx'},'File Selector')
fullpathname=strcat(pathname,filename)
myData=xlsread(fullpathname);
[baris1]=size(myData);
x=myData(1:baris1,1);                %split into I (1st
coloumn) and trip data (2nd coloumn)
y=myData(1:baris1,2);

set(handles.uitable_trainingdata,'data',myData);

% Display Training Data
% -----
axes(handles.axes_TCC);
hLine =loglog(x,y,'-b');
%legend('Training Data');
xlabel('Current (Amp)');
ylabel('Trip time (sec)');
grid on
hold all

% hObject    handle to PB_LOAD (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% --- Executes on button press in PB_LOAD2.
function PB_LOAD2_Callback(hObject, eventdata, handles)
global myData2;
global a;

```

```

global b;

[filename pathname]=uigetfile({'*.xlsx'}, 'File Selector')
fullpathname=strcat(pathname,filename)
myData2=xlsread(fullpathname);
[baris2]=size(myData2);
a=myData2(1:baris2,1);
b=myData2(1:baris2,2);

% Display Training Data 2
% -----
axes(handles.axes_TCC);
loglog(a,b, '-m');
%legend('R-ACB-2');
xlabel('Current (Amp)');
ylabel('Trip time (sec)');
grid on
hold on

% hObject      handle to PB_LOAD2 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see
GUIDATA)

% --- Executes on button press in PB_START.
function PB_START_Callback(hObject, eventdata, handles)
global myData;
global x;
global y;

[baris1]=size(myData);
x=myData(1:baris1,1);
y=myData(1:baris1,2);

ipu=81;
itest1=myData(1:baris1,1);

%LAGRANGE
for k=1:baris1 %jumlah data test
    for i=1:baris1
        pembilang(i)=itest1(k)-x(i);
    end
    for i=1:baris1
        for j=1:baris1
            penyebut(j,i)=x(i)-x(j);
        end
    end
end

```

```

end
for i=1:baris1
    penyebut(i,i)=1;
end
for j=1:baris1
    atas=1;
    bawah=1;
    for i=1:baris1
        if(i==j)
            atas=atas;
        else
            atas=atas*pembilang(i);
            bawah=bawah*penyebut(i,j);
        end
    end
    atas=atas;
    bawah=bawah;
    poli(j)=y(j)*atas/bawah;
end
hasil1(k)=poli(1)+poli(2);
for i=3:baris1
    hasil1(k)=hasil1(k)+poli(i);
end
end

tabel(:,1)=x;
tabel(:,2)=hasil1(:,:);
set(handles.uitable_lagrange,'data',tabel);

% Display Lagrange result
% -----
axes(handles.axes_TCC);
cla
loglog(x,hasil1,'r',x,y,'-b');
%legend('R-ACB-2');
legend('Lagrange Output','IEC');
xlabel('Current (Amp)');
ylabel('Trip time (sec)');
grid on
hold all

% hObject      handle to PB_START (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see
GUIDATA)

```



```

% --- Executes on button press in PB_SAVE_tr.
function PB_SAVE_tr_Callback(hObject, eventdata, handles)

[filename,pathname] = uiputfile('*.xlsx','Save as');
%col=get(handles.uitable_trainingdata,'columnname');
data=get(handles.uitable_trainingdata,'data');
num=[data];
xlswrite([pathname filename],num);

% hObject      handle to PB_SAVE_tr (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see
GUIDATA)

% --- Executes on button press in PB_SAVE_lgr.
function PB_SAVE_lgr_Callback(hObject, eventdata, handles)

[filename,pathname] = uiputfile('*.xlsx','Save as');
%col=get(handles.uitable_trainingdata,'columnname');
data=get(handles.uitable_lagrange,'data');
num=[data];
xlswrite([pathname filename],num);

% hObject      handle to PB_SAVE_lgr (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see
GUIDATA)

% --- Executes on button press in pb_clear_data.
function pb_clear_data_Callback(hObject, eventdata, handles)
global myData
global tabel

myData=0;
tabel=0;
set(handles.uitable_trainingdata,'data',myData);
set(handles.uitable_lagrange,'data',tabel);

% hObject      handle to pb_clear_data (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see
GUIDATA)

```

```

% --- Executes on button press in PB_TEST.
function PB_TEST_Callback(hObject, eventdata, handles)
global myData;
global x;
global y;

[baris1]=size(myData);
x=myData(1:baris1,1);
y=myData(1:baris1,2);

ipu=81;
test1=str2num(get(handles.edit_arus_test,'string'))
ref1=(0.14*0.65)/((test1/ipu)^0.02-1);
%ref1=(-1.05*log10(test1))+1.03);
set(handles.edit_waktu_test_tr,'string', ref1);

%LAGRANGE
for i=1:baris1
pembilang(i)=test1-x(i);
end

for i=1:baris1
    for j=1:baris1
        penyebut(j,i)=x(i)-x(j);
    end
end

for i=1:baris1
    penyebut(i,i)=1;
end

for j=1:baris1
    atas=1;
    bawah=1;
    for i=1:baris1
        if(i==j)
            atas=atas;
        else
            atas=atas*pembilang(i);
            bawah=bawah*penyebut(i,j);
        end
    end

    atas=atas;
    bawah=bawah;

```

```

poli(j)=y(j)*atas/bawah;
end
hasiltest1=poli(1)+poli(2);
for i=3:barisl
hasiltest1=hasiltest1+poli(i);
end
Err1=(abs(hasiltest1-ref1)/ref1)*100;

set(handles.edit_waktu_test_lgr,'string', hasiltest1);
set(handles.edit_err_test,'string', Err1);

% Display Test result
% -----
axes(handles.axes_TCC);
loglog(test1,hasiltest1,'+k');
%legend('Testing');
grid on
%hold on

% hObject      handle to PB_TEST (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see
GUIDATA)

% --- Executes on button press in PB_Clear_test.
function PB_Clear_test_Callback(hObject, eventdata, handles)
axes(handles.axes_TCC);
grid on
cla

ref1=0;
hasiltest1=0;
Err1=0;
set(handles.edit_arus_test,'string','');
set(handles.edit_waktu_test_tr,'string', ref1);
set(handles.edit_waktu_test_lgr,'string', hasiltest1);
set(handles.edit_err_test,'string', Err1);

% hObject      handle to PB_Clear_test (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see
GUIDATA)

```

```

% --- Executes on button press in pb_load_data_sample.
function pb_load_data_sample_Callback(hObject, eventdata,
handles)
global myData3;
global c;
global d;

[filename pathname]=uigetfile({'*.xlsx'}, 'File Selector')
fullpathname=strcat(pathname, filename)
myData3=xlsread(fullpathname);
[baris3]=size(myData3);
c=myData3(1:baris3,1);
d=myData3(1:baris3,2);
set(handles.uitable_sample, 'data', myData3);

% hObject      handle to pb_load_data_sample (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see
GUIDATA)

% --- Executes on button press in pb_test_sample.
function pb_test_sample_Callback(hObject, eventdata,
handles)
global myData;
global x;
global y;
global myData3;
global c;
global d;

[baris1, kolom1]=size(myData);
x=myData(1:baris1,1);
y=myData(1:baris1,2);

[baris3, kolom3]=size(myData3);
c=myData3(1:baris3,1);
d=myData3(1:baris3,2);

%ipu=81;
itest1(1:baris3,1)=myData3(1:baris3,1);

%LAGRANGE
for z=1:baris3
for i=1:baris1

```

```

        for a=1:baris1
            pembilang(i)=itest1(z,1)-x(i);
        end
    end

    for i=1:baris1
        for j=1:baris1
            penyebut(j,i)=x(i)-x(j);
        end
    end

    for i=1:baris1
        penyebut(i,i)=1;
    end

    for j=1:baris1
        atas=1;
        bawah=1;
        for i=1:baris1
            if (i==j)
                atas=atas;
            else
                atas=atas*pembilang(i);
                bawah=bawah*penyebut(i,j);
            end
        end

        atas=atas;
        bawah=bawah;
        poli(j)=y(j)*atas/bawah;
    end
    hasiltest1=poli(1)+poli(2);
    for i=3:baris1
        hasiltest1=hasiltest1+poli(i);
    end
    hasiltest2(z,1)=hasiltest1
end
for e=1:baris3
    Err1(e,1)=abs(hasiltest2(e,1)-d(e))/d(e)*100;
end
tabel2(:,1)=c;
tabel2(:,2)=d(:,:);
tabel2(:,3)=hasiltest2(:,1);
tabel2(:,4)=Err1(:,1);
set(handles.uitable_sample, 'data', tabel2);
%save lag.mat
% Display Test result
% -----
cla

```

```

axes(handles.axes_TCC);
loglog(c,d,'-b',c,hasiltest2,'.r');
%legend('Testing');
grid on
hold on

% hObject      handle to pb_test_sample (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see
GUIDATA)

% --- Executes on button press in pb_save_sample.
function pb_save_sample_Callback(hObject, eventdata,
handles)
[filename,pathname] = uiputfile('*.xlsx','Save as');
%col=get(handles.uitable_trainingdata,'columnname');
data=get(handles.uitable_sample,'data');
num=[data];
xlswrite([pathname filename],num);

% hObject      handle to pb_save_sample (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see
GUIDATA)

% --- Executes on button press in pb_clear_sample.
function pb_clear_sample_Callback(hObject, eventdata,
handles)
global tabel2
tabel2=0;
set(handles.uitable_sample,'data',tabel2);
% hObject      handle to pb_clear_sample (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see
GUIDATA)

% --- Executes on button press in pb_hand.
function pb_hand_Callback(hObject, eventdata, handles)
global hLine
set(hLine,'ButtonDownFcn','disp(''This executes'')');

```

```

set(hLine,'Tag','DoNotIgnore');
p = pan;
set(p,'ButtonDownFilter',@mycallback);
set(p,'Enable','on');

% mouse click on the line
function [flag] = mycallback(obj,event_obj)
% If the tag of the object is 'DoNotIgnore', then return
true.
% Indicate what the target is
disp(['Clicked ' get(obj,'Type') ' object'])
objTag = get(obj,'Tag');
if strcmpi(objTag,'DoNotIgnore')
    flag = true;
else
    flag = false;
end

% hObject    handle to pb_hand (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% --- Executes on button press in pb_zoom.
function pb_zoom_Callback(hObject, eventdata, handles)
global hLine
set(hLine,'ButtonDownFcn','disp(''This executes'')');
set(hLine,'Tag','DoNotIgnore');
z = zoom;
set(z,'ButtonDownFilter',@mycallback);
set(z,'Enable','on');
% mouse click on the line
%
function [flag] = mycallback(obj,event_obj)
% If the tag of the object is 'DoNotIgnore', then return
true.
objTag = get(obj,'Tag');
if strcmpi(objTag,'DoNotIgnore')
    flag = true;
else
    flag = false;
end

% hObject    handle to pb_zoom (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB

```

```

% handles      structure with handles and user data (see
GUIDATA)

% --- Executes on button press in pb_pin.
function pb_pin_Callback(hObject, eventdata, handles)
%global tabel;
for n=1:inf
%n=Inf;
axes(handles.axes_TCC);
    [x,y,but] = ginput(1);
    loglog(x,y,'.k');
%for b=1:n
tabel(n,1)=x(1,1);
tabel(n,2)=y(1,1);
set(handles.uitable_trainingdata,'data',tabel);
end
%axes(handles.axes_TCC);
%loglog(tabel(:,1),tabel(:,2),'.k');
% hObject      handle to pb_pin (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see
GUIDATA)

function edit_arus_test_Callback(hObject, eventdata,
handles)
% hObject      handle to edit_arus_test (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit_arus_test as text
%           str2double(get(hObject,'String')) returns contents
of edit_arus_test as a double

% --- Executes during object creation, after setting all
properties.
function edit_arus_test_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to edit_arus_test (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

```



```

% Hint: edit controls usually have a white background on
Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit_waktu_test_tr_Callback(hObject, eventdata,
handles)
% hObject      handle to edit_waktu_test_tr (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit_waktu_test_tr as text
%      str2double(get(hObject,'String')) returns contents
of edit_waktu_test_tr as a double

% --- Executes during object creation, after setting all
properties.
function edit_waktu_test_tr_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to edit_waktu_test_tr (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit_waktu_test_lgr_Callback(hObject, eventdata,
handles)
% hObject      handle to edit_waktu_test_lgr (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit_waktu_test_lgr as text
% str2double(get(hObject,'String')) returns contents
of edit_waktu_test_lgr as a double

% --- Executes during object creation, after setting all
properties.
function edit_waktu_test_lgr_CreateFcn(hObject, eventdata,
handles)
% hObject handle to edit_waktu_test_lgr (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit_err_test_Callback(hObject, eventdata, handles)
% hObject handle to edit_err_test (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit_err_test as text
% str2double(get(hObject,'String')) returns contents
of edit_err_test as a double

% --- Executes during object creation, after setting all
properties.
function edit_err_test_CreateFcn(hObject, eventdata,
handles)

```

```

% hObject    handle to edit_err_test (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit_parameter_Callback(hObject, eventdata,
handles)
% hObject    handle to edit_parameter (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit_parameter as text
%         str2double(get(hObject,'String')) returns contents
of edit_parameter as a double

% --- Executes during object creation, after setting all
properties.
function edit_parameter_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to edit_parameter (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

Halaman ini sengaja di kosongkan

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengujian dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa :

1. Pada pengujian pemodelan kurva standar, jika semakin besar jumlah data input maka tingkat keakuratan semakin baik, seperti pada tabel 4.7.
2. Pada pengujian pemodelan kurva standar jumlah data dengan tingkat keakuratan paling baik adalah 27 dengan nilai error terkecil sebesar 0,070191, seperti hasil yang ditunjukkan pada gambar 4.6.
3. Pemodelan kurva karakteristik non-standar dapat dilakukan dengan membuat titik-titik data yang baru dan dapat diterapkan pada ETAP seperti hasil pada gambar 4.11.
4. Pada pengujian kurva non-standar pada ETAP 12.6, dengan nilai arus beban sebesar 657,7 A, waktu tripnya adalah 1,71 detik lebih cepat dibandingkan waktu trip pada kurva standar yaitu 2,13 detik seperti pada gambar 4.10 dan gambar 4.11.
5. Selain mempercepat waktu trip, pemodelan kurva standar menjadi kurva non-standar juga dapat menyelesaikan permasalahan tentang tumpang tindih antar kurva, seperti yang ditunjukkan pada gambar 4.11.
6. Pemodelan kurva non-standar tidak ditentukan oleh persamaan lagi untuk menemukan waktu trip, karena kurva didesain oleh user dengan cara membuat titik-titik data baru.
7. Pemodelan kurva karakteristik non-standar dapat dilakukan dengan membuat titik-titik data yang baru dan dapat diterapkan pada ETAP.
8. Hasil pengujian prototipe rele arus lebih digital menunjukkan hasil yang mendekati dengan data target pada ETAP dengan rata-rata error = 1.753 %.

5.2 Saran

Saran yang dapat diberikan berdasarkan penelitian ini adalah perlu dilakukan penelitian lanjutan mengenai pemodelan dengan berbagai jenis kurva karakteristik inverse lainnya baik yang standar maupun non-standar. Dan dapat dilakukan dengan mencoba menggunakan metode Artificial Intelligence seperti Neural Network (NN).

DAFTAR PUSTAKA

- [1] Moch Harun Arrosyid, Pemodelan Kurva Karakteristik Inverse Over Current Relay oleh User Menggunakan Adaptive Neuro Fuzzy Inference System, Tugas Akhir Teknik Elektro PENS, Surabaya, 2011.
- [2] Anang Tjahjono, Dimas Okky Anggriawan, Ardyono Priyadi, Margo Pujiانتارا and Mauridhi Hery Purnomo, Overcurrent Relay Curve Modeling and Its Application in the Real Industrial Power Systems Using Adaptive Neuro Fuzzy Inference System, ITS, Surabaya, 2015.
- [3] Anang Tjahjono, Ardyono Priyadi, Margo Pujiانتارا and Mauridhi Hery Purnomo, Overcurrent Relay Curve Modeling Using Adaptive Neuro Fuzzy Inference System, ITS, Surabaya, 2014.
- [4] “ Studi Rele Pengaman (Over Current Relay dan Ground Fault Relay) Pada Pemakaian Distribusi Daya Sendiri Dari PLTU Rembang” Tugas Akhir Teknik Elektro ITS 2013.
- [5] Ir R.Wahyudi. “Diktat Kuliah Pengaman Sistem Tenaga Listrik”, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya.
- [6] Anderson, P.M, “Power System Protection”, John Wiley & Sons, Inc., Canada, Ch. 3, 1998
- [7] “Studi Koordinasi Proteksi Rele Arus Lebih dengan Metode Karakteristik Tripping Non-Standar pada Jaringan 150kV dan 20kV PT.PLN (Persero) APJ Gilimanuk” Tugas Akhir Teknik Elektro ITS 2013.
- [8] Dr. Eng. Supriyanto, M.Sc. “Interpolasi Lagrange”, Lab. Komputer, Departemen Fisika, Universitas Indonesia.

Halaman ini sengaja di kosongkan

RIWAYAT HIDUP



Nama lengkap Nurio Herlambang, lahir pada 16 Oktober 1991 di kota Palembang, Sumatera Selatan. Memulai pendidikan sekolah dasar pada tahun 1998-2004 di SDN 1 Karang Maritim, Bandar Lampung. Melanjutkan studi sekolah menengah di SMPN 23 Bandar Lampung pada tahun 2004-2007, kemudian praktikan melanjutkan studi di SMA Negeri 1 Bandar Lampung di tahun 2007. Pada tahun 2010 praktikan mengambil studi Diploma 3 di Jurusan Teknik Elektro, Bidang Teknik Telekomunikasi, Politeknik Negeri

Bandung, Jawa Barat. Dan melanjutkan studi Lintas Jalur (S1) Jurusan Teknik Elektro, Bidang Studi Teknik Sistem Tenaga, Institut Teknologi Sepuluh Nopember, Surabaya pada tahun 2014. Semasa perkuliahan penulis aktif dalam seminar dan pelatihan. Email : nurioherlambang@gmail.com